

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ  
Заведующий кафедрой

\_\_\_\_\_ О. В. Непомнящий  
подпись

«\_\_\_\_\_» \_\_\_\_\_ 2021 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 – «Информатика и вычислительная техника»

Система для учета связей с работодателями

Руководитель

\_\_\_\_\_  
подпись, дата

ст. преподаватель

К. В. Пушкарев

Выпускник

\_\_\_\_\_  
подпись, дата

С. А. Панкратов

Нормоконтролер

\_\_\_\_\_  
подпись, дата

К. В. Пушкарев

Красноярск 2021

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ  
Заведующий кафедрой

\_\_\_\_\_ О. В. Непомнящий  
подпись

«\_\_\_\_\_» \_\_\_\_\_ 2021 г.

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ  
в форме бакалаврской работы**

Студенту Панкратову Семену Александровичу.

Группа: КИ17-08Б. Направление (специальность): 09.03.01

«Информатика и вычислительная техника».

Тема выпускной квалификационной работы: «Система для учёта связей с работодателями».

Утверждена приказом по университету № 5631/с от 27.04.2021.

Руководитель ВКР: Пушкарев К. В., старший преподаватель кафедры ВТ ИКИТ СФУ.

Исходные данные для ВКР: нет.

Перечень разделов ВКР:

1. Анализ задания на выпускную квалификационную работу.
2. Проектирование и реализация системы.
3. Инструкции.

Перечень графического материала: не требуется.

## 1 Задание

Разработать систему для учёта связей кафедры высшего учебного заведения с работодателями.

## 2 Основные требования

Клиентская часть системы должна быть кроссплатформенной для настольных систем (ОС Windows, Linux). Система в целом должна иметь открытый исходный код и обладать следующими возможностями:

а) ввод/вывод данных через графический интерфейс, а также в машиночитаемом файловом формате;

б) поддержка многопользовательской работы;

в) поиск информации по различным критериям;

г) разграничение прав доступа пользователей к системе;

д) создание отчётов.

Хранимая и обрабатываемая информация:

а) работодатели: название, область деятельности, решаемые задачи из области информатики и вычислительной техники и т. д.;

б) тестовые задания для студентов от работодателей;

в) проводимые мероприятия с работодателями;

г) связи с образовательной программой: какие результаты обучения требуются работодателю;

д) практики: номер семестра в учебном плане, работодатель, руководители, студенты, результаты.

Руководитель ВКР

\_\_\_\_\_

подпись

К. В. Пушкарев

Задание принял к исполнению

\_\_\_\_\_

подпись

С. А. Панкратов

31 октября 2020 г.

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Система для учёта связей с работодателями» содержит 56 страниц текстового документа, 2 таблицы, 40 иллюстраций, 12 использованных источников, 1 приложение.

ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗА ДАННЫХ, РАБОТОДАТЕЛЬ, КАФЕДРА, QT, C++, ГРАФИЧЕСКИЙ ИНТЕРФЕЙС, POSTGRESQL.

Цель работы: разработать систему для учёта связей кафедры высшего учебного заведения с работодателями.

Для достижения цели в работе решаются следующие задачи:

- анализ задания на выпускную квалификационную работу;
- проектирование;
- реализация системы.

Кафедра и предприятия заинтересованы в том, чтобы подготовка специалистов отвечала потребностям предприятий. В связи с этим необходима система, которая могла бы автоматизировать хранение и поиск информации о связях с работодателями.

В результате анализа задания на выпускную квалификационную работу были рассмотрены существующие решения, также был проведён обзор инструментов. В результате проектирования системы, была разработана архитектура, составлена диаграмма прецедентов, построена модель базы данных. Также были составлены и описаны классы для графического интерфейса и для работ с базой данных.

В итоге была спроектирована и реализована система для учёта связей с работодателями.

# СОДЕРЖАНИЕ

Введение.....	4
1 Анализ задания на выпускную квалификационную работу .....	5
1.1 Обзор существующих решений.....	5
1.1.1 Информационная система «Вуз – работодатель».....	6
1.1.2 Информационная система для подбора персонала.....	7
1.1.3 Система информационного взаимодействия вузов, работодателей и студентов.....	7
1.1.4 Итоги обзора.....	8
1.2 Обзор инструментов .....	9
1.3 Итоги анализа задания.....	11
2 Проектирование и реализация системы.....	11
2.1 Архитектура системы .....	11
2.2 Схема потоков окон .....	12
2.3 Диаграмма прецедентов .....	17
2.4 Модель базы данных.....	21
2.5 Описание классов.....	25
2.6 Графический интерфейс .....	26
3 Инструкции .....	33
3.1 Инструкция пользователя.....	33
3.1.1 Вход в систему .....	33
3.1.2 Главное окно.....	33
3.1.3 Работодатель.....	34
3.1.4 Практики .....	37
3.1.5 Мероприятия.....	42
3.1.6 Образовательные программы .....	43

3.1.7 Работники предприятия.....	46
3.1.8 Сотрудники университета.....	47
3.1.9 Студенты.....	48
3.2 Инструкция для администратора.....	49
3.3 Инструкция для разработчика.....	50
Заключение.....	53
Список использованных источников.....	54
ПРИЛОЖЕНИЕ А (обязательное) Логическая модель базы данных.....	56

## ВВЕДЕНИЕ

В процессе обучения студенты проходят практику на предприятиях, а после выпуска становятся их работниками. И кафедра, и работодатели заинтересованы в том, чтобы подготовка специалистов отвечала потребностям предприятий. В связи с этим необходима система, которая могла бы автоматизировать хранение и поиск информации о связях с работодателями: требуемых результатах обучения, проводимых практиках, мероприятиях, тестовых заданиях.

Цель работы: разработать систему для учёта связей кафедры высшего учебного заведения с работодателями.

Клиентская часть системы должна быть кроссплатформенной для настольных систем (ОС Windows, Linux). Система в целом должна иметь открытый исходный код и обладать следующими возможностями:

- ввод/вывод данных через графический интерфейс, а также в машиночитаемом файловом формате;
- поддержка многопользовательской работы;
- поиск информации по различным критериям;
- разграничение прав доступа пользователей к системе;
- создание отчётов.

Для достижения цели в работе решаются следующие задачи:

- анализ задания на выпускную квалификационную работу;
- проектирование;
- реализация системы.

## 1 Анализ задания на выпускную квалификационную работу

В соответствии с заданием необходимо разработать информационную систему для учёта связей с работодателями под операционные системы Windows, Linux. Для выполнения задания, был выполнен анализ аналогов, при этом отмечены их достоинства и недостатки. В ходе анализа не было обнаружено готовых решений с открытым исходным кодом, отвечающих требованиям задания.

### 1.1 Обзор существующих решений

На сегодняшний день существуют несколько информационных систем, которые служат для учета связи с работодателями.

В таблице 1 представлены некоторые информационные системы, направленные на учет связи с работодателями.

Таблица 1 – Основные сведения о информационных системах

Название	Учебные и тестовые задания	Хранение истории проведенных практик	Анкетирование	Создание отчетов
Информационная система «Вуз – работодатель»	+	-	+	-
Информационная система для подбора персонала	-	-	+	-
Система информационного взаимодействия вузов, работодателей и студентов	-	-	-	-



### **1.1.1 Информационная система «Вуз – работодатель»**

«Вуз — работодатель» [1] — информационная система, которая служит инструментом разрешения противоречий между характеристиками выпускников и ожиданием рынка труда. В данной ИС реализована возможность создания конкурсов, курсов и семинаров для студентов от работодателей. После создания курса, семинара или конкурса все студенты, подходящие под критерии работодателя, будут оповещены об этом. Кроме того, у студента присутствует функция просмотра всех работодателей и их требований к студентам. Также в данной информационной системе у сотрудника вуза присутствует возможность создания анкет для работодателя, сбора статистики и просмотра отзывов от работодателей.

Достоинства данной информационной системы:

- хранение и обработка образовательных программ вуза;
- присутствуют учебные и тестовые задания от работодателя;
- возможность обмена сообщениями;
- работодатели могут оценивать работу студентов посредством отзывов;
- поиск результатов по различным критериям.

Недостатки данной информационной системы:

- отсутствие машиночитаемого ввода/вывода информации;
- отсутствие возможности создания отчетов;
- отсутствие возможности хранить и обрабатывать информацию о историях практик.

### **1.1.2 Информационная система для подбора персонала**

Информационная система для подбора персонала позволяет руководителям предприятий решить проблему подбора персонала на вакантные должности, ведь это является довольно трудоемким и сложным процессом, начиная с изучения резюме и заканчивая приемом на работу [2]. В данной ИС для обеспечения надежности и объективности результатов используют различные анкеты и опросники, а для их обработки используют метод свертки критериев [2, с. 84]. При этом учитываются и ответы соискателя, и установленная заранее степень важности вопроса. Весовой коэффициент вопроса может быть определён, например, методом ранжирования или парных сравнений [2, с. 84] вне информационной системы.

Достоинства данной информационной системы:

- обработка анкет происходит автоматически с помощью математических методов;

- поиск результатов по различным критериям.

Недостатки данной информационной системы:

- отсутствие учебных и тестовых заданий от работодателя;
- необходимо привлекать экспертов для установки оценки значимости;
- отсутствие машиночитаемого ввода/вывода информации;
- отсутствие хранения и обработки образовательных программ вуза;
- отсутствие возможности хранить и обрабатывать информацию о истории практик;
- отсутствие возможности создания отчетов.

### **1.1.3 Система информационного взаимодействия вузов, работодателей и студентов**

Система информационного взаимодействия вузов, работодателей и студентов [3] – система, позволяющая университетам адаптивно управлять

индивидуальными образовательными траекториями, учитывая интересы работодателей и студентов, с помощью сбора, хранения, систематизации информации от работодателей.

Достоинства данной информационной системы:

- присутствует связь требований работодателя с образовательной программой;
- возможности хранения и обработки образовательных программ вуза;
- поиск результатов по различным критериям;
- возможность системы контролировать образовательную траекторию, учитывая интересы работодателя и студентов;
- поиск результатов по различным критериям.

Недостатки данной информационной системы:

- отсутствие учебных и тестовых заданий от работодателя;
- отсутствие возможности хранить и обрабатывать информацию о истории практик;
- отсутствие машиночитаемого ввода/вывода информации;
- отсутствие возможности создания отчетов.

#### **1.1.4 Итоги обзора**

Из обзора решений можно сделать вывод, что все рассмотренные аналоги предназначены для хранения и обработки информации о взаимодействии студентов и работодателей в контексте образовательной программы вуза.

Из трех информационных систем взаимодействие между вузом и работодателем лучше всего происходит в информационной системе «Вуз – Работодатель» посредством отзывов и возможности предоставить информацию об образовательных программах вуза работодателю. Реализация системы анкетирования, представленная в «Информационной системе для подбора персонала» должна ускорить и автоматизировать выбор работников на рабочие места.

Проанализировав информационные системы, можно сделать вывод, о том, что ни одна из них не обладает следующими функциями:

- ввод данных через машиночитаемый файловый формат;
- создание отчетов;
- возможность хранить и обрабатывать информацию о истории практик.

## 1.2 Обзор инструментов

Информационная система должна работать на платформах Windows, Linux с использованием графического интерфейса пользователя. Для реализации этого существуют следующие языки программирования и фреймворки:

- Java с инструментом JavaFX;
- C++ с фреймворком Qt;
- Python с фреймворком Qt.

Java – объектно-ориентированный язык программирования общего назначения [4]. JavaFX – инструмент для создания графических интерфейсов под Java. Может использоваться для создания настольных и интернет приложений.

C++ – компилируемый язык программирования общего назначения с уклоном в сторону системного программирования. Поддерживает такие парадигмы программирования, как процедурное программирование, модульное программирование, абстракция данных, объектно-ориентированное программирование, обобщённое программирование [5].

Qt – кроссплатформенный фреймворк для разработки ПО на различных языках программирования [6]. Написанный код можно компилировать для Linux, Windows и других операционных систем. Qt имеет удобное IDE Qt Creator. Она включает в себя Qt Designer для создания графического интерфейса.

Python – интерпретируемый, высокоуровневый язык программирования. Python поддерживает фреймворк Qt, который позволяет ему использовать графические приложения [7].

Qt включает в себя множество библиотек, которые упростят разработку информационной системы. Код компилируется как на Windows, так и на Linux. Также Qt включает в свой состав графический фреймворк. Исходя из это в качестве фреймворка для графического интерфейса был выбран Qt.

В результате анализа вышеперечисленных языков программирования, был сделан вывод о том, что C++ является компилируемым языком программирования, что дает ускорение в работе программы. В итоге, так как C++ является самым производительным из вышеперечисленных, выбор был сделан в пользу C++ и Qt.

Для хранения различной информации в ИС необходимо использовать базу данных, наиболее популярные:

- SQLite;
- MySQL;
- PostgreSQL.

SQLite — это реляционная база данных без сервера SQL [8]. SQLite не имеет сервера, а хранит всю базу данных на устройстве, где установлена программа, тем самым делая невозможным многопользовательское использования приложения.

MySQL — это реляционная база данных [9]. MySQL используется в качестве сервера, к которому обращаются клиенты, однако существуют варианты использования MySQL автономно.

PostgreSQL — свободная система управления базами данных. В отличие от других СУБД, поддерживает объектно-ориентированный подход к БД. PostgreSQL поддерживает NoSQL [10].

Для организации многопользовательской ИС необходимо использовать MySQL или PostgreSQL, поскольку SQLite не имеет сервера, а является

библиотекой для приложений, которая позволяет читать и записывать файлы, используя SQL запросы.

PostgreSQL является более многофункциональной СУБД, чем MySQL и благодаря своим технологиям, способна быстрее обрабатывать сложные запросы. Исходя из этих преимуществ, в качестве БД была выбрана PostgreSQL.

### 1.3 Итоги анализа задания

В результате обзора аналогов был сделан вывод, что ни одна из рассмотренных информационных систем не обладает следующими функциями:

- ввод данных через машиночитаемый файловый формат;
- создание отчетов;
- возможности хранить и обрабатывать информацию о практиках.

В качестве языка программирования был выбран C++ с библиотекой Qt. В качестве базы данных была выбрана PostgreSQL.

## 2 Проектирование и реализация системы

### 2.1 Архитектура системы

На рисунке 1 представлена схема архитектуры системы. Пользователь взаимодействует с клиентом, производит вход в систему. После входа, пользователь попадает в систему, где уже может взаимодействовать с данными, хранящимися в удалённой базе данных PostgreSQL.



Рисунок 1 – Архитектура системы

## 2.2 Схема потоков окон

На рисунках 2-5 представлены схемы потоков окон системы.

Для добавления или изменения студентов в главном меню необходимо выбрать пункт меню «Студенты», в появившемся диалоговом окне, нажав кнопку «Добавить» можно добавить нового студента, заполнив окно «Создать студента», либо изменить студента, сделав двойное нажатие на студента из списка. Для загрузки студента из файла формата CSV, необходимо в меню «Студенты» нажать на кнопку «Загрузить», появится диалоговое окно для загрузки студентов.

Для добавления или изменения практики в главном меню необходимо нажать кнопку «Добавить», либо сделать двойное нажатие на практику из списка соответственно. В диалоговом окне «Практика» или в окне «Изменить практику», при нажатии на кнопку «Добавить» откроется диалоговое окно «Прохождение практики». При нажатии на кнопку «Изменить», откроется окно поиска. При нажатии на кнопку «Добавить», откроется диалоговое окно для добавления отчета. Для добавления несколько прохождений практики, необходимо нажать на кнопку «Добавить несколько», откроется диалоговое окно для добавления нескольких прохождений практики.

Для создания отчета о проведенных практиках, необходимо из главного меню нажать на кнопку «Отчет», откроется диалоговое окно для создания отчета.

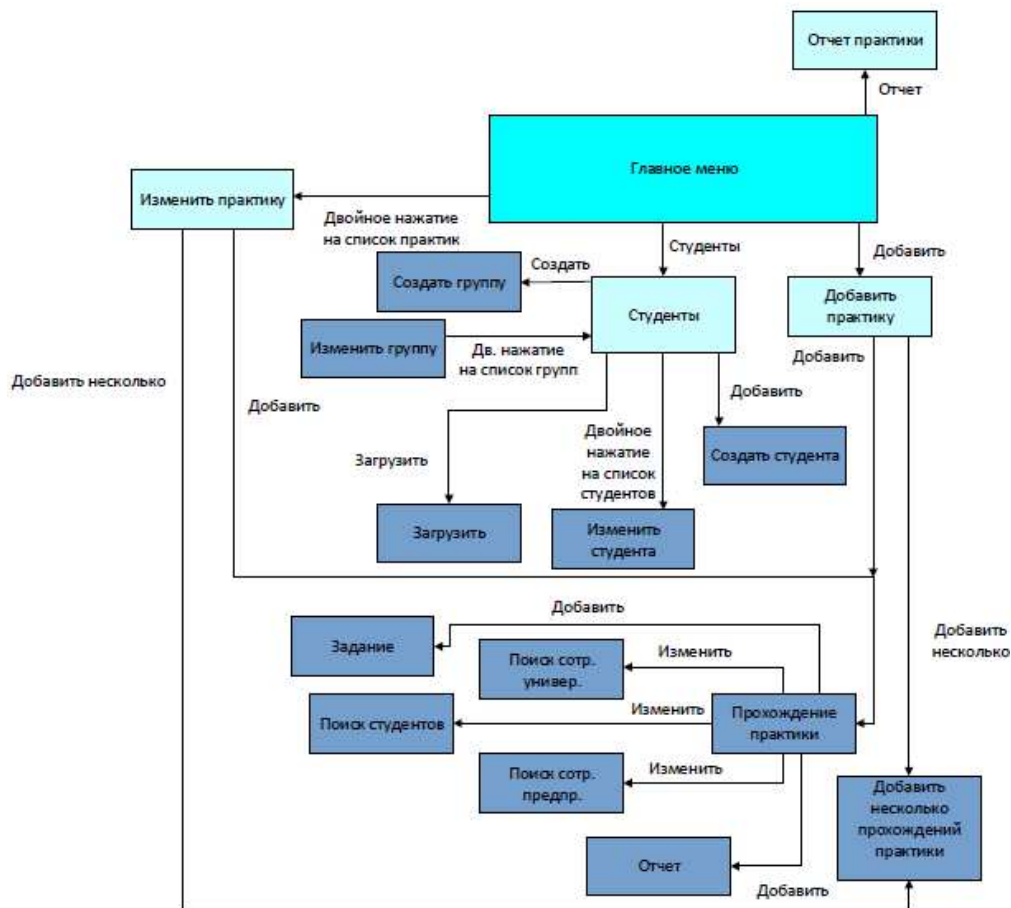


Рисунок 2 – Схема потоков окон практик и студентов

Для добавления или изменения мероприятий в главном меню необходимо выбрать пункт меню «Мероприятия», в диалоговом окне, нажав кнопку «Добавить» можно добавить новое мероприятие, заполнив окно «Добавить мероприятие», либо изменить мероприятие, сделав двойное нажатие на мероприятие из списка. Для добавления или изменения сотрудников университета в главном меню необходимо выбрать пункт меню «Сотрудники университета», в открывшемся диалоговом окне, нажав на кнопку «Добавить» можно добавить нового сотрудника университета, заполнив окно «Добавить сотрудника университета», либо изменить сотрудника университета, сделав двойное нажатие на сотрудника университета из списка. Также в диалоговом окне «Сотрудники университета» можно добавить или изменить кафедру, нажав на кнопку «Добавить», либо сделав двойное нажатие на кафедру из списка соответственно.



Для создания отчета о мероприятиях необходимо из окна «Мероприятия» нажать на кнопку «Отчет», откроется диалоговое окно для создания отчета.

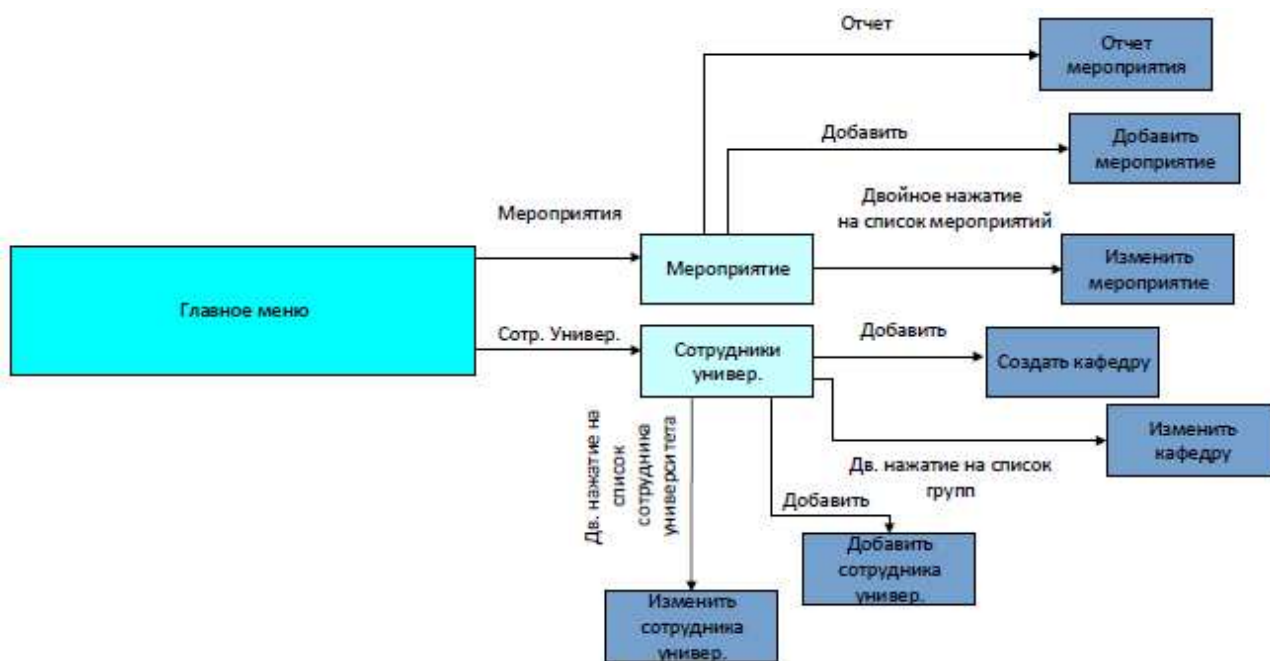


Рисунок 3 – Схема потоков окон мероприятий и сотрудников университета

Для добавления или изменения работодателей в главном меню необходимо нажать кнопку «Добавить», либо сделать двойное нажатие на работодателя из списка. В диалоговом окне «Добавить работодателя» или «Изменить работодателя», нажав на кнопку «Добавить» откроется диалоговое окно «Добавить контактное лицо». В нем, заполнив данные, можно добавить контактное лицо работодателя. Также в диалоговом окне «Добавить работодателя» или «Изменить работодателя», нажав на кнопку «Добавить» откроется диалоговое окно «Добавить руководителя практики». В нем, заполнив данные, можно добавить руководителя практики. Для изменения контактного лица или руководителя практики, в окне «Добавить работодателя» или «Изменить работодателя», необходимо сделать двойное нажатие на руководителя практики или контактное лицо из списка.

Аналогичным методом можно добавлять или изменять руководителей практик, либо контактных лиц из окна «Работники предприятия», для перехода в это окно из главного меню необходимо выбрать пункт работники предприятия. Также в диалоговом окне «Изменить работодателя» или «Добавить работодателя», можно добавлять или изменять дополнительную информацию о работодателе (решаемые задачи, направления сотрудничества, область деятельности), нажав на кнопку «Дополнительно».

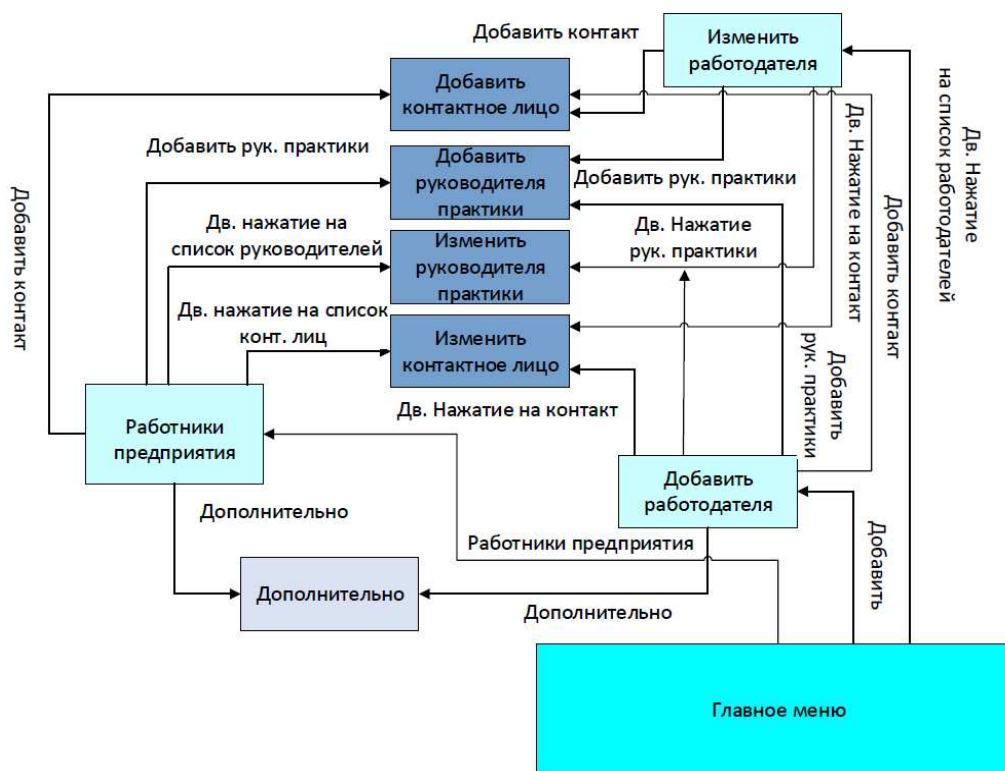


Рисунок 4 – Схема потоков окон работодателя и работников предприятия

Для добавления или изменения образовательных программ в главном меню необходимо выбрать пункт меню «Образовательные программы», в котором, нажав кнопку «Добавить» можно добавить новую образовательную программу, заполнив окно «Добавить образовательную программу», либо изменить образовательную программу, сделав двойное нажатие на образовательную программу из списка.

В диалоговом окне «Образовательные программы», при нажатии на кнопку «Дополнительно», откроется диалоговое окно «Дополнительно», в котором можно добавить дисциплину, заполнив диалоговое окно «Добавить дисциплину». Также из окна «Дополнительно» можно изменить дисциплину, сделав двойное нажатие на дисциплину из списка. Для добавления или изменения образовательного результата из диалоговых окон «Изменить дисциплину», «Добавить дисциплину», «Добавить образовательный результат», «Изменить образовательный результат» необходимо нажать на кнопку «Результат».

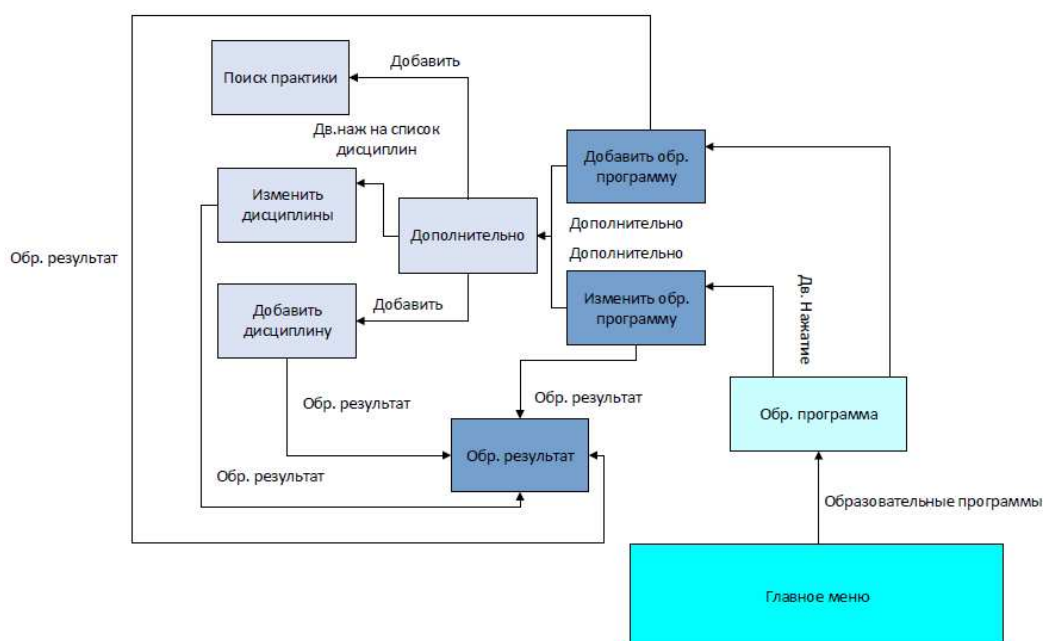


Рисунок 5 – Схема потоков окон образовательных программ и дисциплин

## 2.3 Диаграмма прецедентов

На рисунке 6 представлена диаграмма прецедентов.

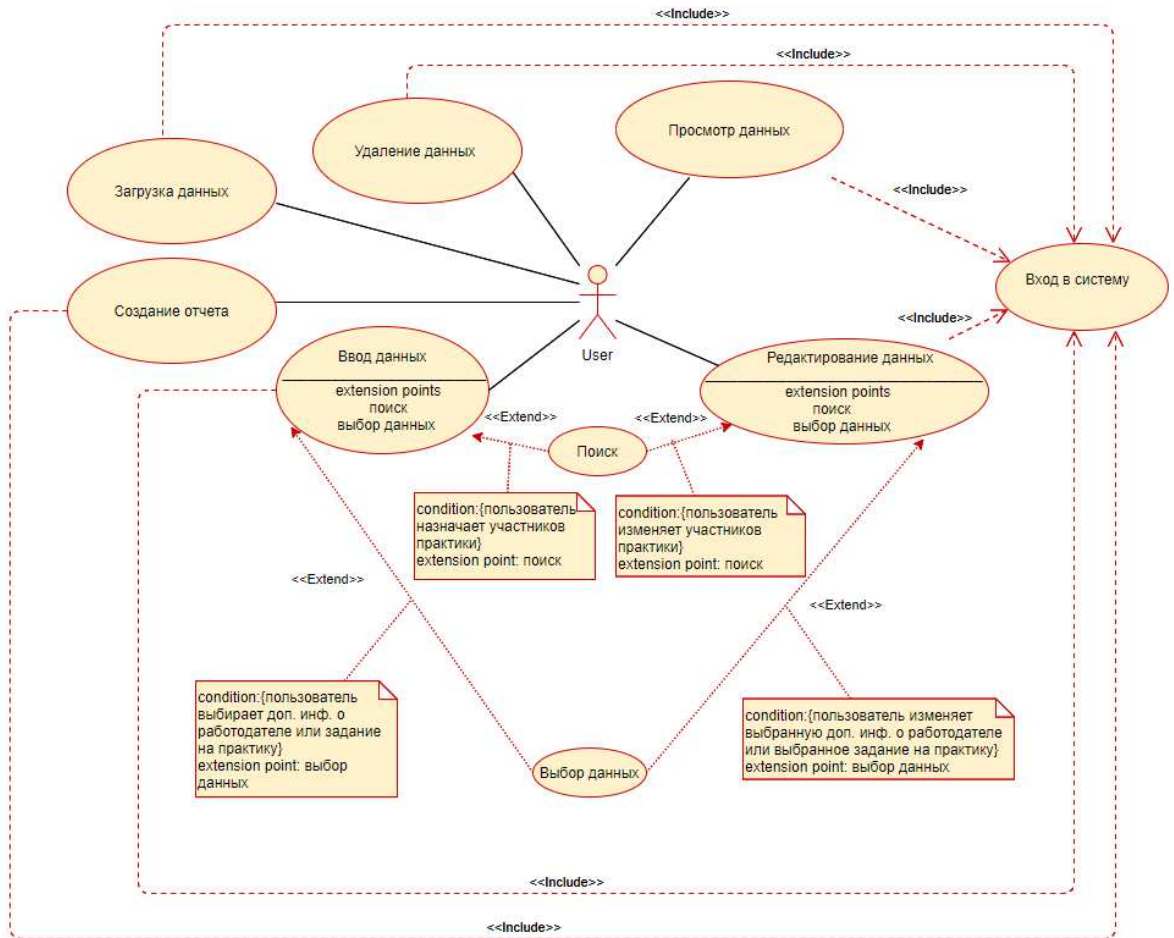


Рисунок 6 – Диаграмма прецедентов

Текстовое описание прецедентов представлено ниже.

**Название прецедента:** «Ввод данных».

Цель сценария: добавление данных в базу данных.

Предусловие: пользователь находится в окне «Студенты», либо в окне «Мероприятия», либо в окне «Практики», либо в окне «Сотрудники университета», либо в окне «Дополнительно» и нажимает кнопку «Добавить».

Основной сценарий:

A. Вывод окна ввода данных.

**Точки расширения:** поиск, выбор данных

B. Пользователь вводит данные в окне и нажимает кнопку «Добавить».

C. Происходит добавление данных в список.

D. Происходит синхронизация с базой данных.

Постусловие: пользователь видит добавленные данные.

Условие ввода в действие альтернативных сценариев.

Условие 1. Неправильно заполнено окно ввода данных.

A. Вывод ошибки.

Условие 2. Нажата кнопка «Отмена».

A. Закрытие окна ввода.

**Название прецедента:** «Просмотр данных».

Цель сценария: просмотр данных.

Предусловие: пользователь находится в окне «Студенты», либо в окне «Мероприятия», либо в окне «Практики», либо в окне «Сотрудники университета», либо в окне «Дополнительно» и делает двойное нажатие на элемент из списка.

Основной сценарий:

A. Регистрация выбранного элемента из списка.

B. Вывод информации о выбранном элементе.

**Название прецедента:** «Редактирование данных».

Цель сценария: редактирование.

Предусловие: пользователь находится в окне «Студенты», либо в окне «Мероприятия», либо в окне «Практики», либо в окне «Сотрудники университета», либо в окне «Дополнительно» и делает двойное нажатие на элемент из списка.

Основной сценарий:

A. Получение выбранного элемента из списка.

B. Вывод информации.

**Точки расширения:** поиск, выбор данных.

C. Пользователь изменяет данные и нажимает на кнопку «Изменить».

D. Происходит изменение данных в списке.

E. Происходит синхронизация с базой данных.

Постусловие: пользователь видит измененных данные.

**Название прецедента: «Выбор данных»**

Цель сценария: выбор данных для работодателя (выбор области деятельности, решаемых задач или направлений сотрудничества), либо для прохождения практики (выбор задания).

Предусловие: пользователь находится в окне «Дополнительно», либо в окне «Прохождение практики» и делает двойное нажатие на элемент из списка.

Основной сценарий:

- A. Получение выбранного элемента из списка.
- B. Происходит изменение данных в списке.
- C. Происходит синхронизация с базой данных.
- D. Подсвечивание выбранных данных.

**Название прецедента: «Выход»**

Цель сценария: выход.

Предусловие: пользователь находится в главном меню и нажимает на «Выход» или закрывает главное окно.

Основной сценарий:

- A. Главное окно закрывается.

**Название прецедента: «Удаление данных»**

Цель сценария: удаление данных из базы данных

Предусловие: пользователь находится в окне «Студенты», либо в окне «Мероприятия», либо в окне «Практики», либо в окне «Сотрудники университета», либо в окне «Дополнительно», выбирает данные из списка и нажимает кнопку «Удалить».

Основной сценарий:

- A. Удаление данных из списка.
- B. Происходит синхронизация с базой данных.

Постусловие: пользователь видит удаление данных.

**Название прецедента: «Поиск»**

Цель сценария: поиск студентов, сотрудников предприятия, либо

сотрудников университета в базе данных.

Предусловие: пользователь находится в окне «Поиск студентов», «Поиск сотрудников университета», либо поиск «Сотрудников предприятия».

Основной сценарий:

А. Происходит поиск по базе данных.

Постусловие: пользователь видит результаты поиска в виде списка.

**Название прецедента:** «Вход в систему»

Цель сценария: вход в систему.

Предусловие: пользователь находится в окне входа в систему, вводит логин и пароль и нажимает кнопку «Вход».

Основной сценарий:

А. Данные проверяются на корректность.

Постусловие: пользователь попадает в главное окно.

Условие ввода в действие альтернативных сценариев.

Условие 1. Проблемы с соединением с базой данных

А. Система выводит сообщение об ошибке.

Условие 2. Введены некорректные данные.

А. Система выводит сообщение об ошибке.

**Название прецедента:** «Загрузка данных»

Цель сценария: Загрузка данных из файла формата csv.

Предусловие: пользователь находится в окне «Студенты».

Основной сценарий:

А. Пользователь указывает формат данных csv.

В. Пользователь указывает входной файл.

С. Происходит добавление данных в списки.

Д. Происходит синхронизация с базой данных.

Постусловие: пользователь видит результат загрузки данных.

**Название прецедента:** «Создание отчетов»

Цель сценария: Создание отчетов.

Предусловие: пользователь находится в главном окне или окне

«Мероприятия».

Основной сценарий:

А. Пользователь выбирает данные для отчета.

В. Пользователь указывает директорию для отчета.

Постусловие: создание отчета.

## **2.4 Модель базы данных**

После анализа задания было выделено 17 основных сущностей.

Центральной из них является сущность «Работодатель». Работодатель имеет в штате контактные лица, которые служат только для связи с ним. Также работодатель имеет в штате руководителей практик. Работодатель организует множество практик для университета.

В свою очередь каждая практика имеет свое собственное задание, из базы данных всех заданий. Практику ведут несколько действующих лиц. Одно представляет университет, другое работодателя. У каждой практики имеется множество отчетов, которые составили студенты.

Руководители практик от предприятия имеют работодателя и оценивают множество отчетов.

Студенты проходят множество практик.

Образовательная программа включает в себя множество практик и имеет образовательный результат, который требуется работодателю.

Дисциплина состоит в образовательной программе и имеет образовательный результат.

В данной системе под образовательным результатом понимают не оценку, а полученные навыки в ходе обучения, которые требуются работодателю.

На рисунке 7 представлена концептуальная модель базы данных.



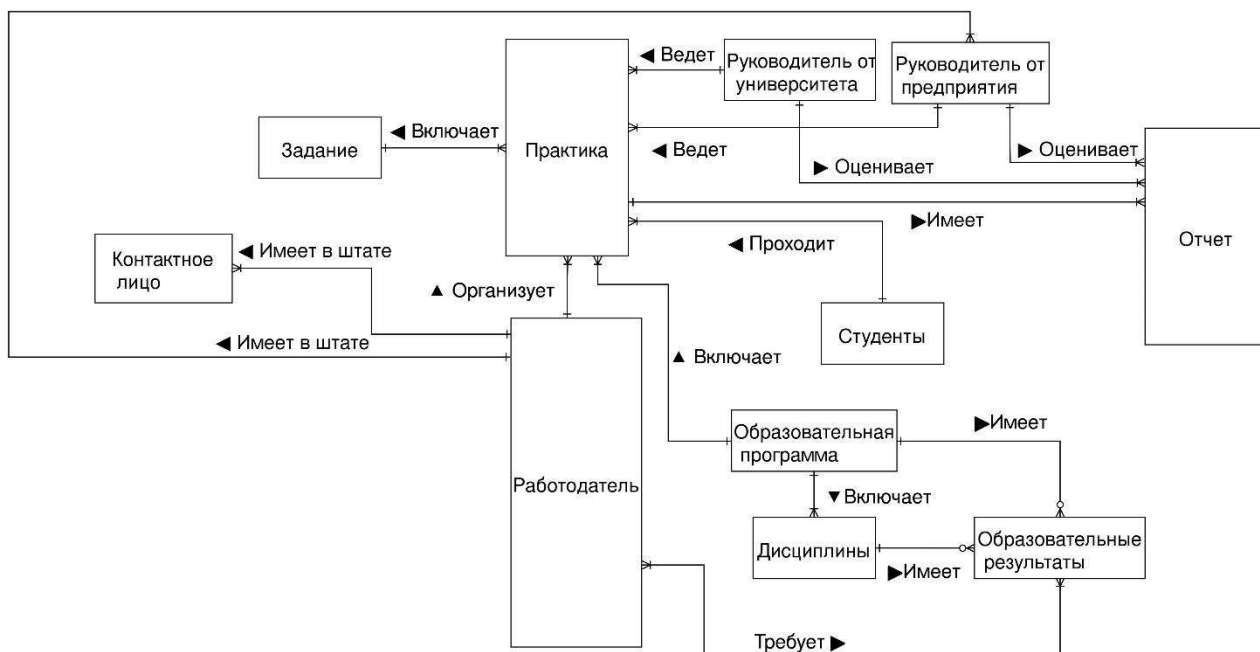


Рисунок 7 – Концептуальная модель базы данных

В приложение А представлена логическая модель базы данных.

В таблице 2 представлено описание сущностей. Сущность «Прохождение практики» является служебной для организации связи между сущностями. «Персона» также является служебной сущностью для хранения персональной информации «Студент», руководителя практики предприятия, руководителя практики университета и контактного лица с работодателями. Сущности «Область деятельности», «Решаемые задачи» и «Направления сотрудничества» необходимы для хранения параметров работодателя. Сущность «Мероприятия» предназначена для хранения мероприятий, сущность «Тестовые задания» хранит тестовые задания, которые необходимо пройти студенту.

Таблица 2 – Описание сущностей

Сущность	Поля и их назначение	Тип данных поля
Работодатель	<b>id</b> – идентификатор	integer
	<b>наименование</b>	text
Прохождение практики	<b>id</b> – идентификатор	integer
	<b>практика_id</b> – связь с работодателем	integer
	<b>рук.пр.предпр. id</b> – связь с рук. практики предприятия	integer
	<b>рук.пр.универ. id</b> – связь с рук. практики университета	integer
	<b>студент_id</b> – связь с студентами	integer
	<b>задание_id</b> – связь с заданием	integer
Мероприятия	<b>id</b> – идентификатор	integer
	<b>наименование</b>	text
	<b>описание</b>	text
	<b>дата начала</b>	date
	<b>дата конца</b>	date
	<b>работодатель_id</b> – связь с работодателем	integer
Тестовые задания	<b>id</b> – идентификатор	integer
	<b>практика_id</b> – связь с практикой	integer
	<b>данные</b> – двоичные данные для хранения тестового задания	data
	<b>id работодателя</b> – связь с работодателем	integer
Направление сотрудничества	<b>id</b> – идентификатор	integer
	<b>наименование</b>	text
	<b>описание</b>	text
Решаемые задачи	<b>id</b> – идентификатор	integer
	<b>наименование</b>	text
	<b>описание</b>	text
Области деятельности	<b>id</b> – идентификатор	integer
	<b>наименование</b>	text
	<b>описание</b>	text
Практики	<b>id</b> – идентификатор	integer
	<b>наименование</b>	text
	<b>обр. программа_id</b> – связь с обр. программой	integer
	<b>начало</b>	date
	<b>конец</b>	date
	<b>работодатель_id</b> – связь с работодателем	integer
	<b>семестр</b>	integer
Задание	<b>id</b> – идентификатор	integer
	<b>наименование</b>	text
	<b>описание</b>	text
Персона	<b>id</b> – идентификатор	integer
	<b>имя</b>	text
	<b>фамилия</b>	text
	<b>отчество</b>	text
	<b>телефон</b>	text
	<b>e-mail</b>	text

Окончание таблицы 2

Сущность	Поля и их назначение	Тип данных поля
Руководитель практики предприятия	<b>id</b> – идентификатор	integer
	<b>работодатель_id</b> – связь с работодателем	integer
	<b>персона_id</b> – связь с персоной	integer
Руководитель практики университет	<b>id</b> – идентификатор	integer
	<b>кафедра_id</b> – связь с работодателем	integer
	<b>персона_id</b> – связь с персоной	integer
Кафедра	<b>id</b> – идентификатор	integer
	<b>наименование</b>	text
Отчет	<b>id</b> – идентификатор	integer
	<b>данные</b> – последовательность байт	byte
	<b>отзыв рук. пр. предпр</b>	text
	<b>отзыв рук. пр. универ</b>	text
	<b>результат практики_id</b> – связь с результатом практики	integer
	<b>рук. пр. предпр_id</b> – связь с рук. по. предпр.	integer
	<b>рук. пр. универ_id</b> – связь с рук. пр. универ (integer)	integer
<b>прох. практики_id</b> – связь с прохождением практики	integer	
Результаты практики	<b>id</b> – идентификатор	integer
	<b>оценка рук. пр. предпр</b>	text
	<b>прохождение пр. id</b> – связь с прохождением практики	integer
	<b>студента_id</b> – связь со студентом	integer
	<b>оценка рук. пр. универ</b>	text
Группы	<b>id</b> – идентификатор	integer
	<b>номер</b>	text
Студенты	<b>id</b> – идентификатор	integer
	<b>группа_id</b> – связь с группой	integer
	<b>обр. программа_id</b> – связь с обр. программой	integer
	<b>персона_id</b> – связь с персоной	integer
Обр. программа	<b>id</b> – идентификатор	integer
	<b>наименование</b>	text
	<b>описание</b>	text
	<b>направление</b>	text
	<b>направленность</b>	text
Дисциплина	<b>id</b> – идентификатор	integer
	<b>семестр</b>	integer
	<b>наименование</b>	text
	<b>описание</b>	text
	<b>обр. программа_id</b> – связь с образовательной программой	integer
Образовательный результат	<b>id</b> – идентификатор	integer
	<b>приобретённые навыки</b>	text
	<b>обр. программа_id</b> – связь с образовательной программой	integer
	<b>дисциплина_id</b> – связь с дисциплиной	integer
Тестовые задания	<b>id</b> – идентификатор	integer
	<b>данные</b> – последовательность байт	byte
	<b>наименование файла</b>	text
	<b>практика_id</b> – связь с практикой	integer
	<b>работодатель_id</b> – связь с работодателем	integer

## 2.5 Описание классов

Для работы с базой данной используется библиотека QxOrm [11]. Эта библиотека построена таким образом, что для каждой сущности в базе данных сопоставляется свой класс, а данные сущностей хранятся в виде атрибутов классов. Исходя из этого, было сформировано 20 классов, которые позволяют взаимодействовать с базой данных.

Центральным классом является класс Employer. Класс Employer имеет атрибут m\_name. Также класс Employer имеет множество атрибутов контейнеров для реализации связи с другими классами (Activity, Collaboration, Task, ContactEmployer, Practice и т.д.). В свою очередь, классы, которые связаны с Employer имеют атрибут класса Employer или атрибуты контейнеры класса Employer, с помощью которого могут взаимодействовать с работодателем или работодателями.

По такому принципу реализованы все классы в данной работе. На рисунке 8 представлена диаграмма класса Employer с некоторыми его связями.

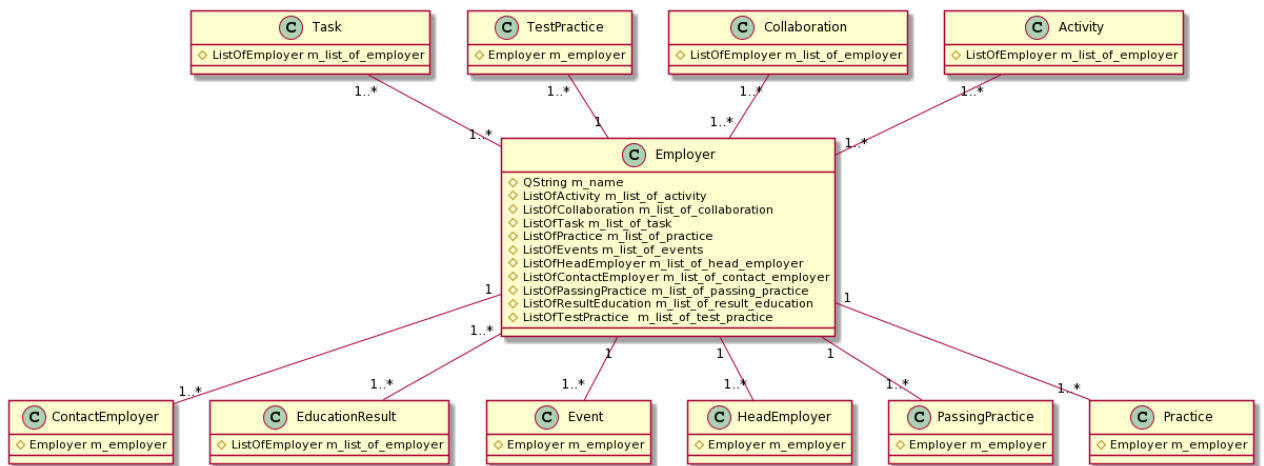


Рисунок 8 – Диаграмма класса Employer

## 2.6 Графический интерфейс

Для реализации графической части был использован фреймворк Qt. На рисунке 9 изображена структура классов, которые используются для реализации графического интерфейса для взаимодействия с контактными лицами, руководителями практик, решаемыми задачами, направлениями сотрудничества и областями деятельности работодателя.

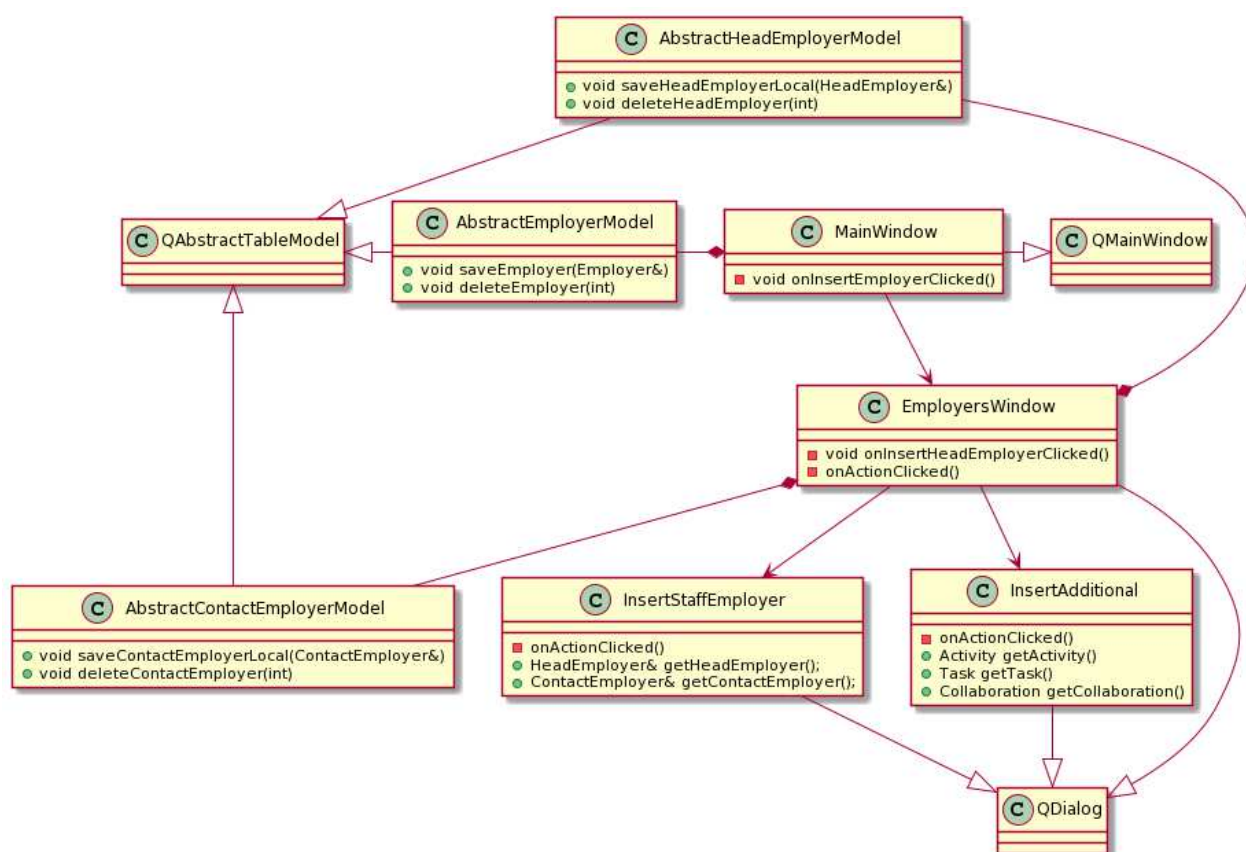


Рисунок 9 – Диаграмма классов, используемых для реализации графического интерфейса для взаимодействия с данными работодателя

Класс `MainWindow` содержит в себе экземпляр класса `AbstractEmployerModel` и `EmployerWindow`. Класс `AbstractEmployerModel` предназначен для сохранения, изменения и удаления работодателей в базе данных. Также этот класс наследует `QAbstractTableModel`, что позволяет ему отображать всех работников в виде таблицы. `EmployerWindow` содержит в

себе экземпляры классов `AbstractContactEmployerModel` и `AbstractHeadEmployerModel`, они служат для сохранения, изменения, удаления, демонстрации контактов работодателей и руководителей практик. `InsertStaffEmployer` — это окно, в которое вводится информация о контакте работодателя или руководителя практики. `InsertAddition` — окно для ввода дополнительной информации о работодателе.

Класс `MainWindow` содержит в себе слот `onInsertEmployerClicked`, который вызывается после нажатия кнопки «Добавить». Данный слот предназначен для вызова окна «Работодатель» и сохранения работодателя.

Класс `AbstractEmployerModel` содержит методы `saveEmployer` для добавления работодателя и `deleteEmployer` для удаления работодателя.

Класс `AbstractHeadEmployerModel` содержит в себе методы `saveHeadEmployerLocal` для добавления руководителя практики и `deleteHeadEmployer` для удаления руководителя практики.

Класс `AbstractContactEmployerModel` содержит в себе методы `saveContactEmployerLocal` для добавления контактного лица и `deleteContactEmployer` для удаления контактного лица.

Класс `InsertStaffEmployer` содержит слот `onActionClicked`, который вызывается при нажатии на кнопку «Добавить». Данный слот предназначен для испускания сигнала `accepted`. Также данный класс содержит методы `getHeadEmployer` для получения руководителя практики и `getContactEmployer` для получения контакта работодателя.

Класс `InsertAddition` содержит слот `onActionClicked`, который вызывается при нажатии на кнопку «Ок». Данный слот предназначен для испускания сигнала `accepted`. Также данный класс содержит методы `getActivity` для получения списка областей деятельности работодателя, `getTask` для получения списка решаемых задач работодателя и `getCollaboration` для получения направлений сотрудничества работодателя.

На рисунке 10 изображена диаграмма классов, которые используются для реализации графического интерфейса для взаимодействия с данными практик.

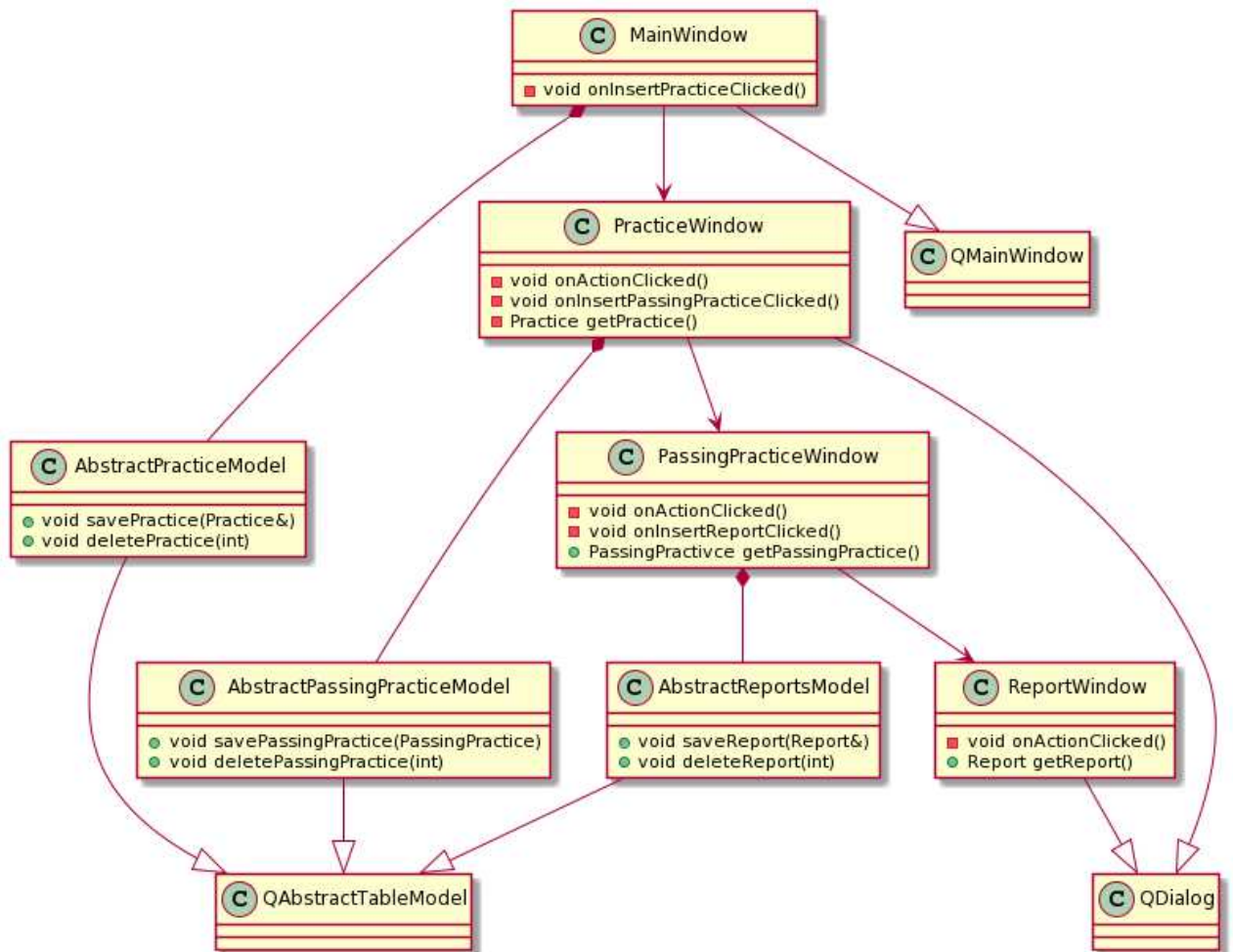


Рисунок 10 – Диаграмма классов, используемых для реализации графического интерфейса для взаимодействия с данными практик

Класс `MainWindow` содержит в себе экземпляры классов `AbstractPracticeModel` и `PracticeWindow`. Класс `AbstractPracticeModel` предназначен для сохранения, изменения и удаления практик в базе данных. Также этот класс наследует `QAbstractTableModel`, что позволяет ему отображать всех практик в виде таблице. `PracticeWindow` содержит в себе экземпляр класса `AbstractPassingPracticeModel`, который служит для сохранения, изменения, удаления, демонстрации прохождения практик.

PassingPracticeWindow — это окно, в который вводится информация о прохождении практик. Оно содержит в себе экземпляр класса AbstractReportModel, который служит для сохранения, изменения, удаления, демонстрации отчетов. ReportWindow — это окно для работы с информацией об отчете.

Класс MainWindow содержит в себе слот onInsertPracticeClicked, который вызывается после нажатия кнопки «Добавить». Данный слот предназначен для вызова окна «Практика» и сохранения практики.

Класс AbstractPracticeModel содержит методы savePractice для сохранения практик и deletePractice для удаления практик.

Класс PracticeWindow содержит в себе слот onActionClicked, который вызывается после нажатия кнопки «Добавить». Данный слот предназначен для испускания сигнала accepted. Также данный класс содержит слот onInsertPassingPracticeClicked, который вызывается после нажатия кнопки «Добавить». Данный слот предназначен для вызова окна «Прохождение практики». Класс PracticeWindow также содержит в себе метод getPractice для получения практики.

Класс AbstractPassingPracticeModel содержит методы savePassingPractice для сохранения прохождения практики и deletePassingPractice для удаления прохождения практики.

Класс PassingPracticeWindow содержит в себе слот onActionClicked, который вызывается после нажатия кнопки «Добавить». Данный слот предназначен для испускания сигнала accepted. Также данный класс содержит слот onInsertReportClicked, который вызывается после нажатия кнопки «Добавить». Данный слот предназначен для вызова окна «Отчет». Класс PracticeWindow также содержит в себе метод getPassingPractice для получения прохождения практики.

Класс AbstractReportModel содержит методы saveReport для сохранения отчетов и deleteReport для удаления отчетов.



Класс ReportWindow содержит в себе слот onActionClicked, который вызывается после нажатия кнопки «Добавить». Данный слот предназначен для испускания сигнала asserpted. Также данный класс содержит метод getReport для получения отчета.

Оставшиеся окна были организованы аналогично.

На рисунке 11 представлено главное окно. На рисунке 12 представлено окно «Работодатель». На рисунке 13 представлено окно «Практики». На рисунке 14 представлено окно «Прохождение практик». На рисунке 15 представлено окно «Отчет».

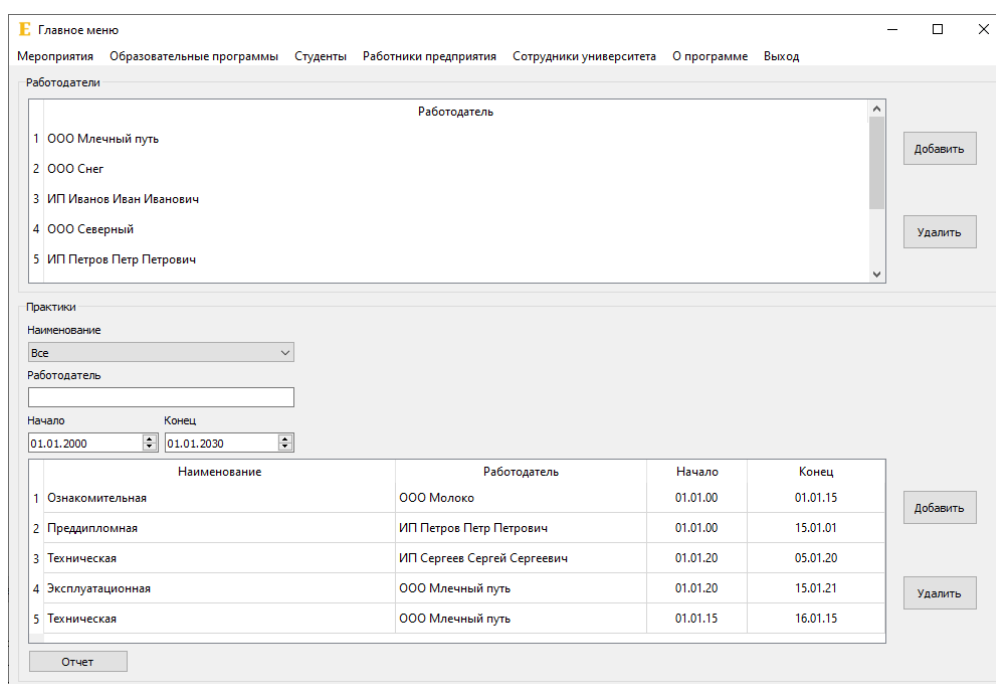


Рисунок 11 – Главное окно

**Работодатель**

Наименование

Руководители практик

Руководитель практики	
1	Егорова Алина Григорьевна
2	Куприянов Артем Артемьевич

Контакты

Контакт	
1	Кудряшова София Владимировна
2	Яшина Евдокия Арёмовна
3	Казакова Таисия Адамовна

Требуемый образовательный результат

Образовательный результат	Источник
1 Программирование	Образовательная программа
2 Изучение информатики	Образовательная программа
3 Оптимизация	Дисциплина

Рисунок 12 – Окно «Работодатель»

**Практика**

Наименование

Работодатель

Идентификатор

Прохождение практик

Студент	

Начало      Конец

Тестовые задания

Рисунок 13 – Окно «Практика»

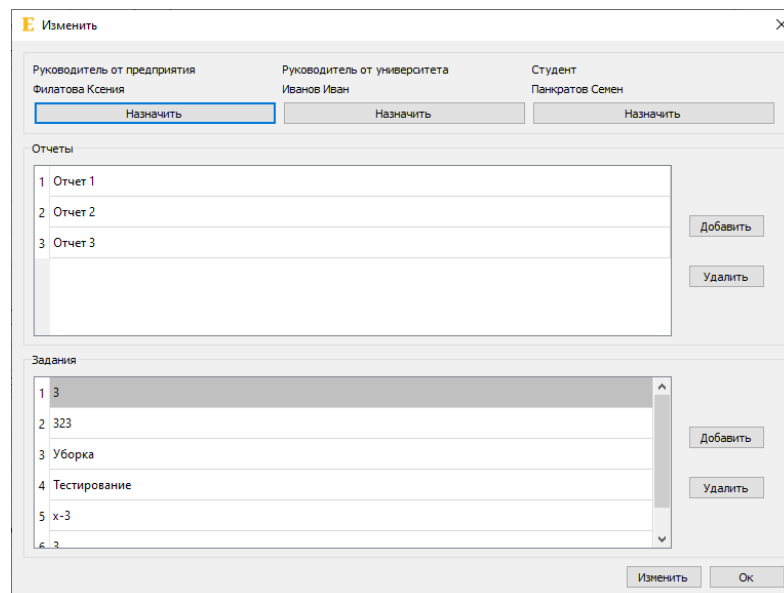


Рисунок 14 – Окно «Прохождение практики»

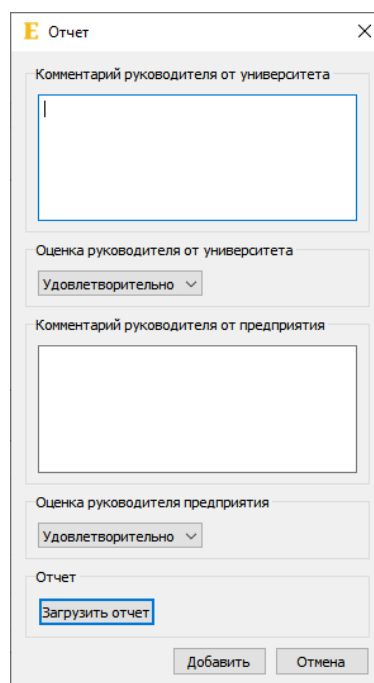


Рисунок 15 – Окно «Отчет»

## 3 Инструкции

### 3.1 Инструкция пользователя

#### 3.1.1 Вход в систему

При открытии системы откроется окно входа в систему. Для входа в систему необходимо ввести логин и пароль. На рисунке 16 представлено окно входа в систему.

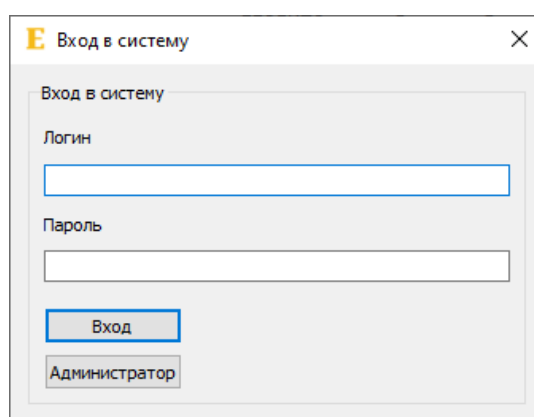


Рисунок 16 – Окно «Вход в систему»

#### 3.1.2 Главное окно

После входа в систему открывается «Главное окно». В главном окне присутствуют пункты меню: «Мероприятия» для работы с мероприятиями, «Образовательные программы» для работы с образовательными программами, «Студенты» для работы со студентами и группами, «Работники предприятия» для работы с сотрудниками всех предприятий, «Сотрудники университета» для работы с сотрудниками университета и кафедрами, «О программе» для просмотра информации о программе, «Выход» для выхода из программы. На рисунке 17 представлено главное окно программы.

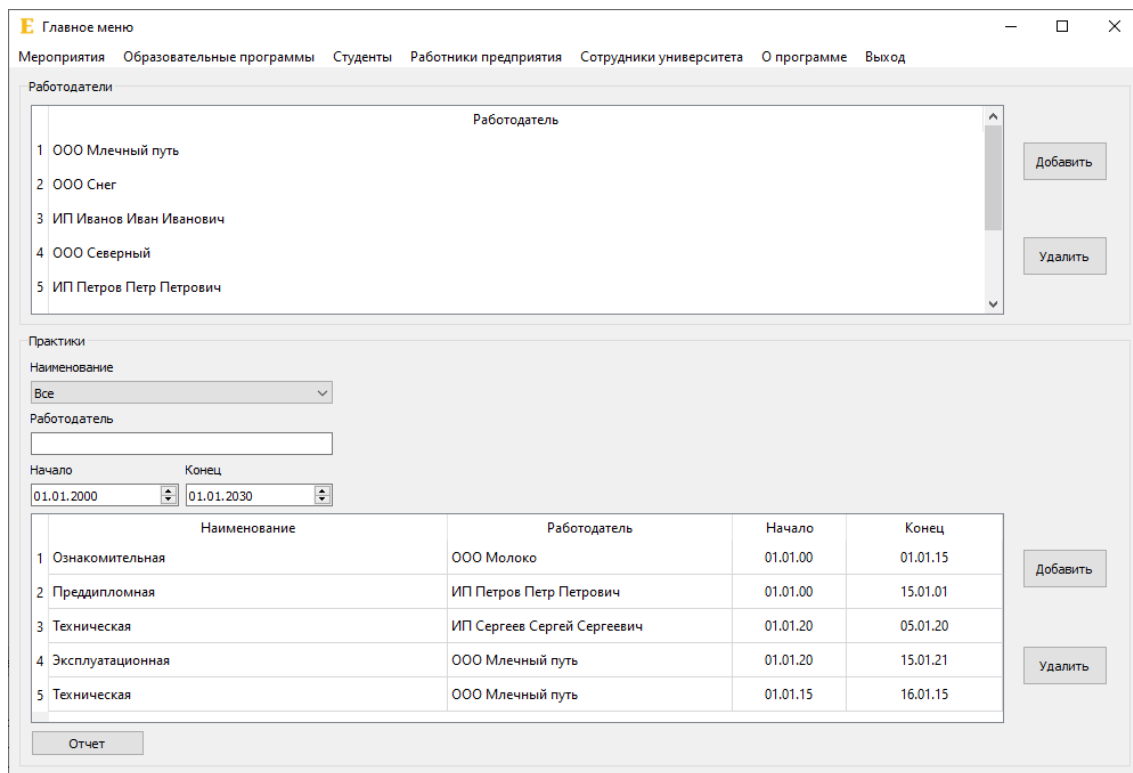


Рисунок 17 – Главное окно

### 3.1.3 Работодатель

Из главного окна можно удалить, добавить, просмотреть или изменить работодателя.

Для просмотра работодателя или изменения работодателя, необходимо сделать двойное нажатие на работодателя из таблицы в главном окне. Появится диалоговое окно, аналогичное рисунку 18.

Для удаления работодателя из главного окна необходимо выбрать работодателя из таблицы и нажать «Удалить».

Для добавления работодателя необходимо нажать на кнопку «Добавить», появится диалоговое окно «Добавить работодателя», рисунок 18.

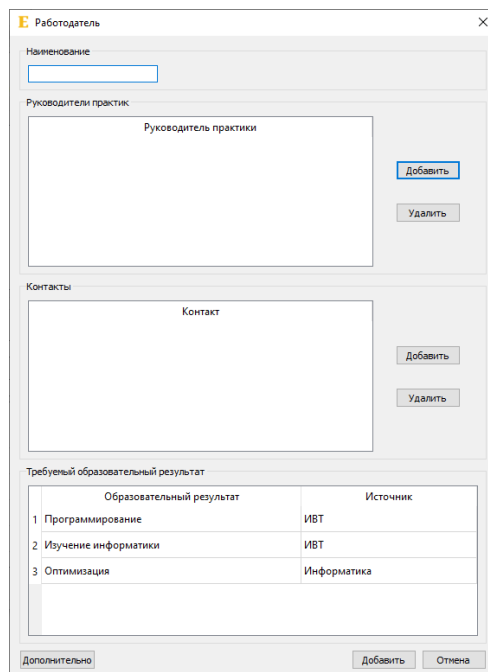


Рисунок 18 – Окно «Работодатель»

В данном окне можно добавить руководителя практики, удалить руководителя практики, просмотреть или изменить руководителя практики. Также в данном окне можно добавить контакт, удалить контакт, просмотреть или изменить контакт. Также можно указать названия работодателя.

Для просмотра или изменения контакта или руководителя практики, необходимо произвести двойное нажатие на него из таблицы.

Для добавления контакта или руководителя практики необходимо нажать на кнопку «Добавить» и заполнить соответствующую форму (рисунок 19).

Рисунок 19 – Окно «Создать работника предприятия»

Для просмотра работника необходимо сделать двойное нажатие на него из списка, появится окно, аналогичное рисунку 19.

В окне «Добавить работодателя» можно выбрать нужный ему образовательный результат, сделав двойное нажатие на образовательной результат из списка, результат будет подсвечен. Аналогичные действие необходимо произвести для удаления требуемого образовательного результата, после этого результат не будет подсвечен.

При нажатии на кнопку «Дополнительно» откроется окно «Дополнительно» для выбора дополнительных параметров работодателя (рисунок 20). Для выбора или удаления параметра необходимо нажать дважды на параметр из таблицы, выбранные параметры подсвечиваются. Для добавления новых параметров необходимо нажать на кнопку «Добавить» и заполнить форму. Для удаления параметра из системы необходимо выделить параметр и нажать на кнопку «Удалить».

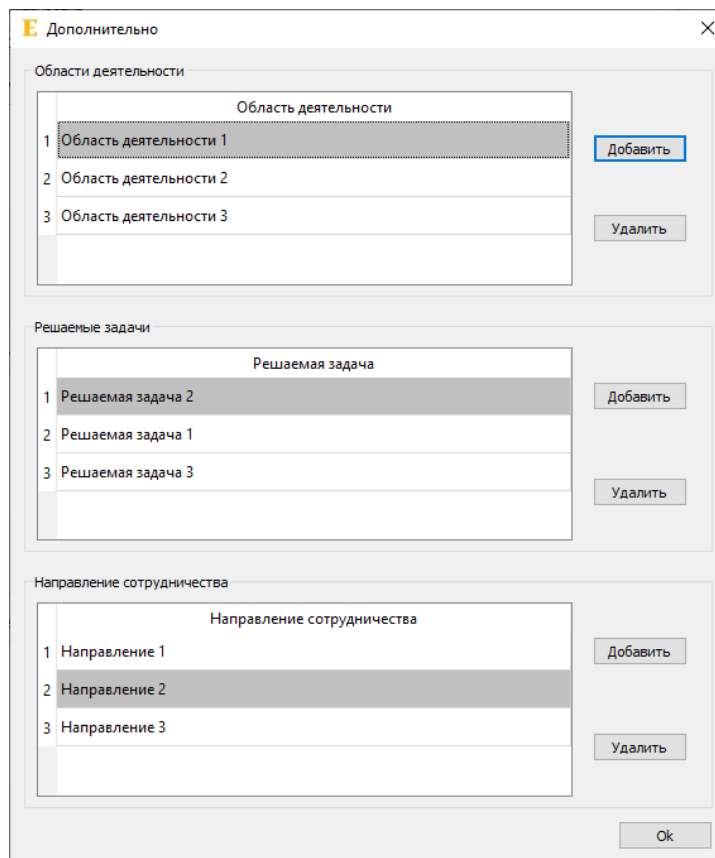


Рисунок 20 – Окно «Дополнительно»

### 3.1.4 Практики

Из главного окна можно удалить, добавить, просмотреть или изменить практику.

Для изменения или просмотра практики из главного окна необходимо сделать двойное нажатие на практику, появится окно, как на рисунке 21.

Для добавления практики необходимо нажать на кнопку «Добавить». Появится окно «Добавить практику» (рисунок 21).



Рисунок 21 – Окно «Практика»

В данном окне можно выбрать работодателя. Задать уникальный индикатор практики. Задать дату начала и конца практики, а также загрузить тестовое задание, которое необходимо пройти студентам. Для дальнейших изменений необходимо нажать на «Создать».

В программе есть возможность назначить нескольких студентов на практику, для этого необходимо нажать на кнопку «Добавить несколько», появится окно «Добавить несколько» (рисунок 22). В данном окне можно выбрать студентов, которых необходимо добавить, также можно выбрать руководителя практики от университета и предприятия, из таблиц. Для завершения добавления необходимо нажать на кнопку «Добавить».

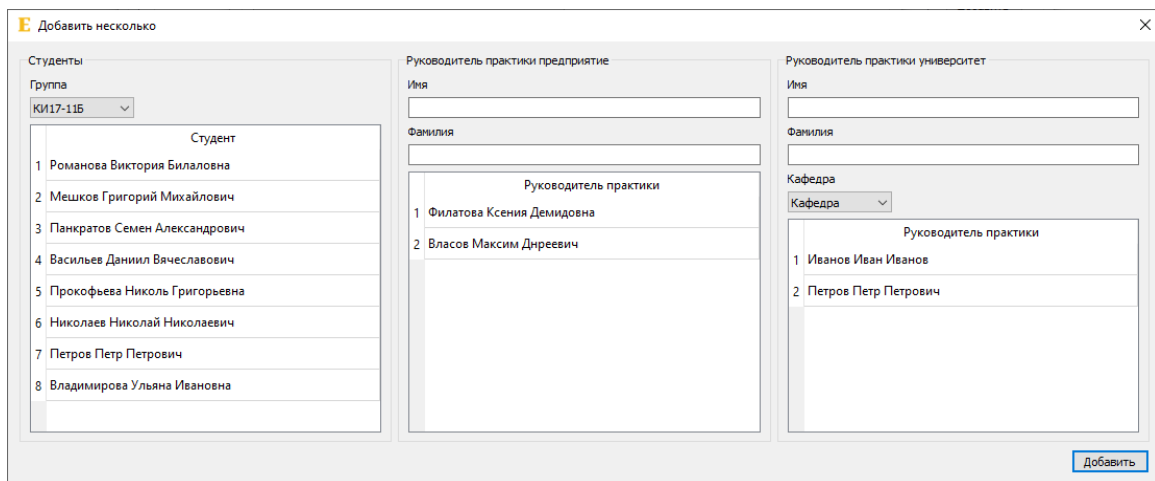


Рисунок 22 – Окно «Добавить несколько»

Из окна «Добавить практику» или «Изменить практику» можно посмотреть прохождение практики, сделав двойное нажатие по таблице, появится диалоговое окно, аналогичное рисунку 23.

Для добавления студентов на практику необходимо нажать на кнопку «Добавить», появится окно «Прохождение практики» (рисунок 23).

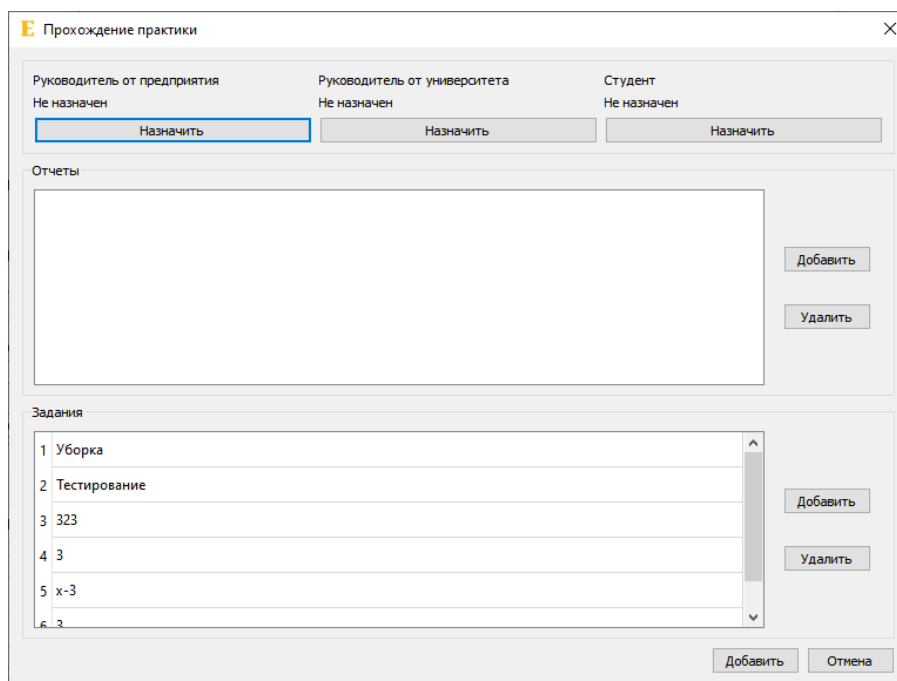
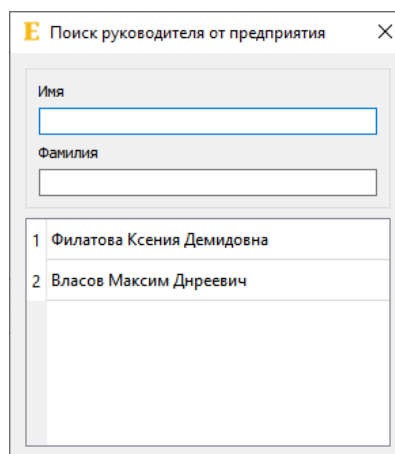


Рисунок 23 – Окно «Прохождение практики»

В данном окне можно назначить руководителя от предприятия, руководителя от университета, студента.

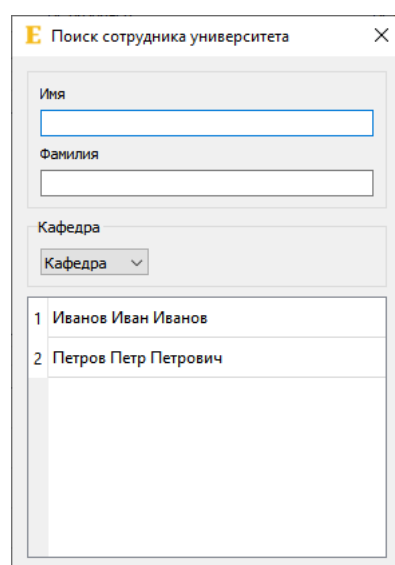
При нажатии на кнопку «Назначить» появится окно поиска участника практики (рисунки 24-26). Для выбора участника практики необходимо сделать двойное нажатие на него из таблицы.



The screenshot shows a window titled «Поиск руководителя от предприятия». It contains two input fields for «Имя» and «Фамилия». Below these fields is a table with two rows of search results:

1	Филатова Ксения Демидовна
2	Власов Максим Днреевич

Рисунок 24 – Окно «Поиск руководителя от предприятия»



The screenshot shows a window titled «Поиск сотрудника университета». It contains two input fields for «Имя» and «Фамилия», and a dropdown menu for «Кафедра». Below these fields is a table with two rows of search results:

1	Иванов Иван Иванов
2	Петров Петр Петрович

Рисунок 25 – Окно «Поиск руководителя от университета»

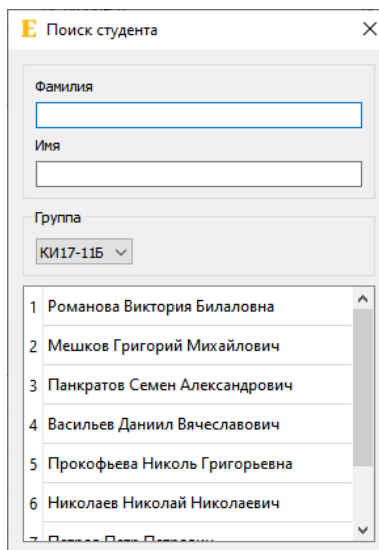


Рисунок 26 – Окно «Поиск студента»

При нажатии на кнопку «Добавить» появится окно для добавления отчета студента, рисунок 27. Нажав на кнопку «Загрузить отчет», можно добавить отчет студента.

Для просмотра или изменения отчета необходимо сделать двойное нажатие на отчет в таблице, появится окно, аналогичное рисунку 27, и, если отчет был загружен в систему, нажав на кнопку «Отчет», можно загрузить отчет.

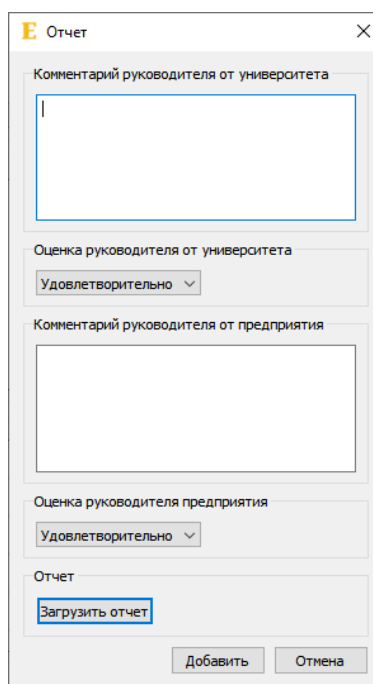


Рисунок 27 – Окно «Отчет»

В окне «Прохождение практики» можно выбрать задание на практику, сделав двойное нажатие на задание из списка, также можно добавить новое задание, нажав на «Добавить» и заполнив форму.

### 3.1.5 Мероприятия

При нажатии из главного окна на пункт меню «Мероприятия» откроется окно мероприятий, в котором отображены все мероприятия (рисунок 28). По нажатию на кнопку «Добавить» откроется окно для добавления мероприятия (рисунок 29).

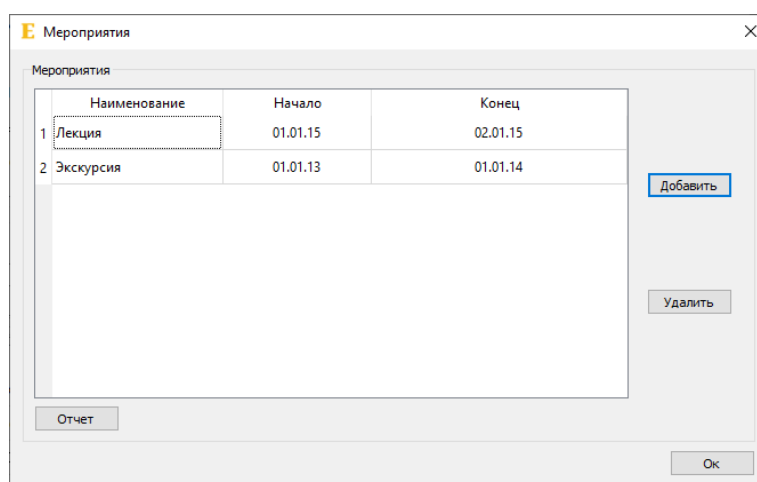


Рисунок 28 – Окно «Мероприятия»

При выделении мероприятия из таблицы и нажатия кнопки «Удалить» произойдет удаление мероприятия.

Для просмотра или изменения мероприятия необходимо сделать двойное нажатие на мероприятия из таблицы, откроется окно, аналогичное рисунку 29. Чтобы изменения мероприятия вступили в силу, необходимо нажать «Изменить».

Мероприятие

Наименование

Описание

Работодатель  
ООО Млечный путь

Начало 01.01.2000      Конец 01.01.2000

Добавить      Отмена

Рисунок 29 – Окно «Мероприятие»

Для создания отчета о мероприятиях необходимо нажать на кнопку «Отчет» и заполнить окно отчета (рисунок 30).

Отчет

Дата

Начало 01.01.2000      Конец 01.01.2000

Сохранить

Рисунок 30 – Окно «Отчет»

### 3.1.6 Образовательные программы

При нажатии из главного окна на пункт меню «Образовательные программы» откроется окно «Образовательные программы» (рисунок 31), в котором отображаются все образовательные программы.

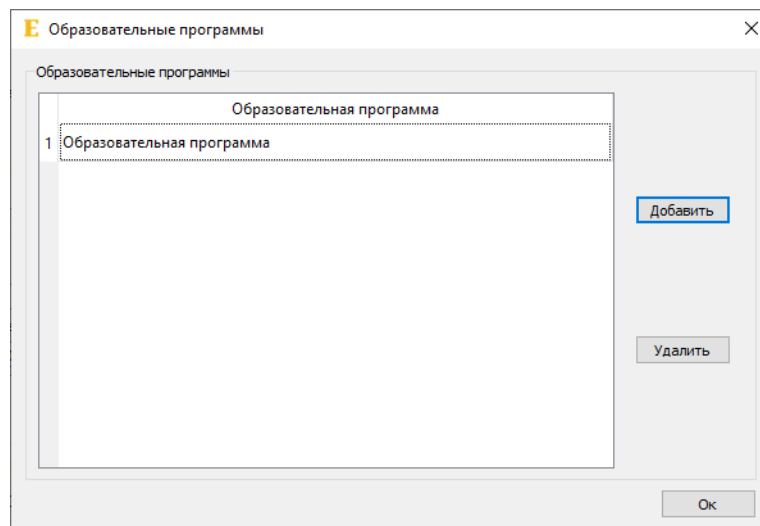


Рисунок 31 – Окно «Образовательные программы»

При выделении образовательной программы из таблицы и нажатия кнопки «Удалить» происходит удаление образовательной программы.

Для добавления образовательной программы необходимо нажать кнопку «Добавить», откроется окно добавления образовательной программы (рисунок 32).

Рисунок 32 – Окно «Образовательная программа»

По нажатию на кнопку «Результаты» откроется окно результатов (рисунок 31), в котором можно просмотреть все результаты образовательной программы, добавить результат, нажав на кнопку «Добавить», просмотреть или изменить результат, сделав двойное нажатие на результат, удалить результат, нажав на него в таблице, а затем нажав на кнопку «Удалить».

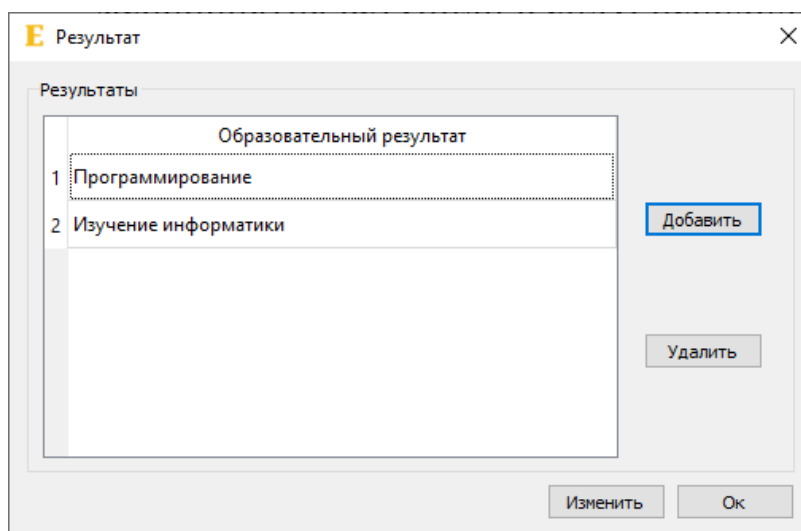


Рисунок 33 – Окно «Результат»

По нажатию на кнопку «Дополнительно» откроется окно «Дополнительно» (рисунок 34), в котором можно просмотреть все дисциплины образовательной программы, удалить их или добавить новую. Также в этом окне можно добавить или удалить практику, окно добавления практики представлено на рисунке 35.

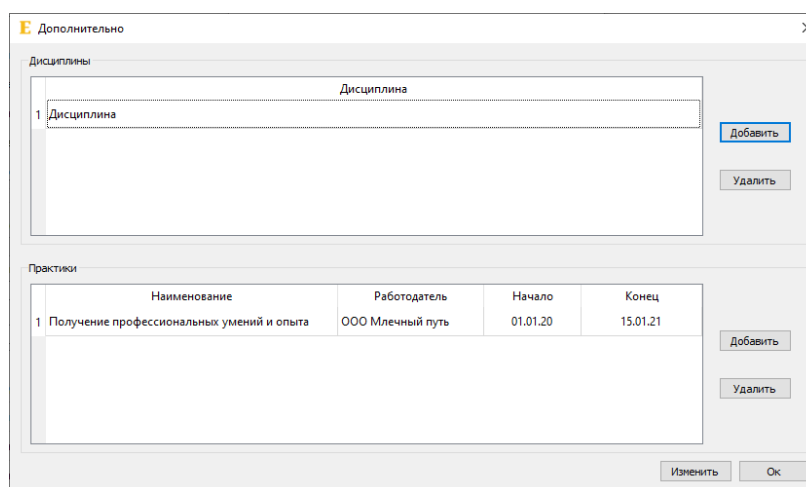


Рисунок 34 – Окно «Дополнительно»



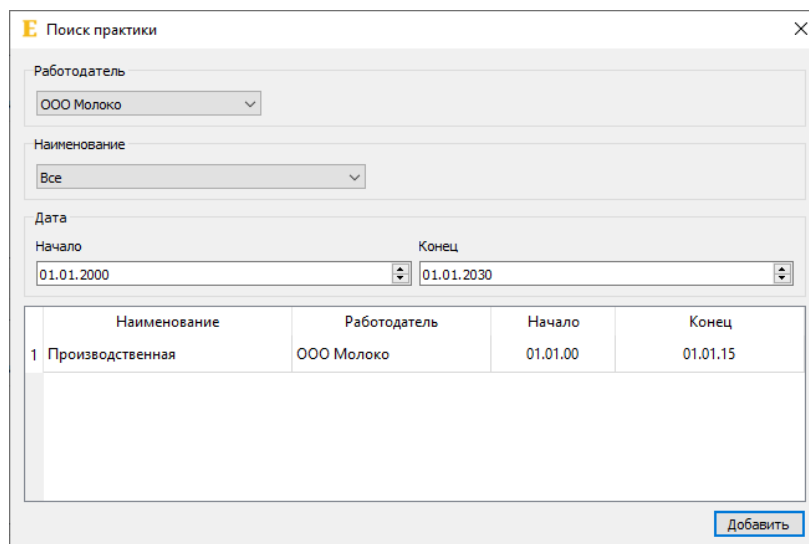


Рисунок 35 – Окно «Поиск практики»

### 3.1.7 Работники предприятия

При нажатии из главного окна на пункт меню «Работники предприятия» откроется окно «Работники предприятия» (рисунок 36), в котором отображаются все руководители практик и контактные лица от всех работодателей.

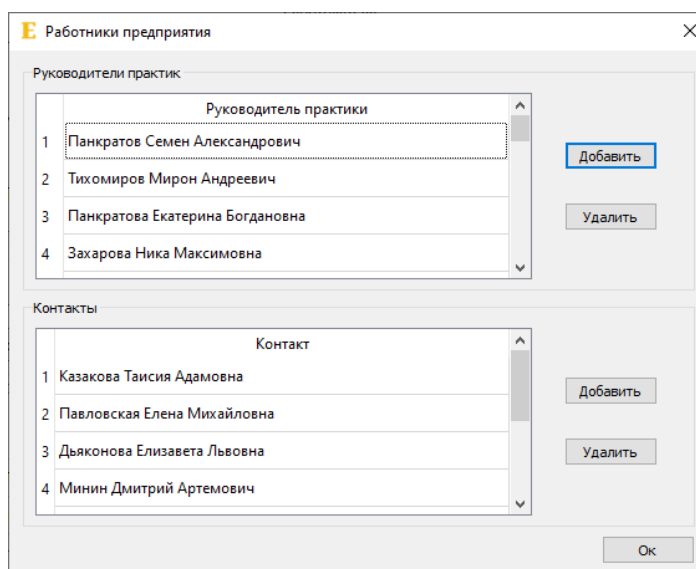


Рисунок 36 – Окна «Работники предприятия»

В данном окне можно добавить руководителя практики, нажав на кнопку «Добавить» и заполнив форму, аналогичную рисунку 19, удалить руководителя практики, выбрав руководителя из таблицы и нажав на «Удалить», просмотреть или изменить руководителя практики, сделав двойное нажатие на руководителя практики из таблицы. Также в данном окне можно добавить контакт, нажав на кнопку «Добавить» и заполнив форму, аналогичную рисунку 19, удалить контакт, выбрав контакт из таблицы и нажав на «Удалить», просмотреть или изменить контакт, сделав двойное нажатие на контакт из таблицы.

### 3.1.8 Сотрудники университета

При нажатии из главного окна на пункт меню «Сотрудники университета» откроется окно «Сотрудники университета» рисунок 37, в котором отображаются все руководители практик и кафедры.

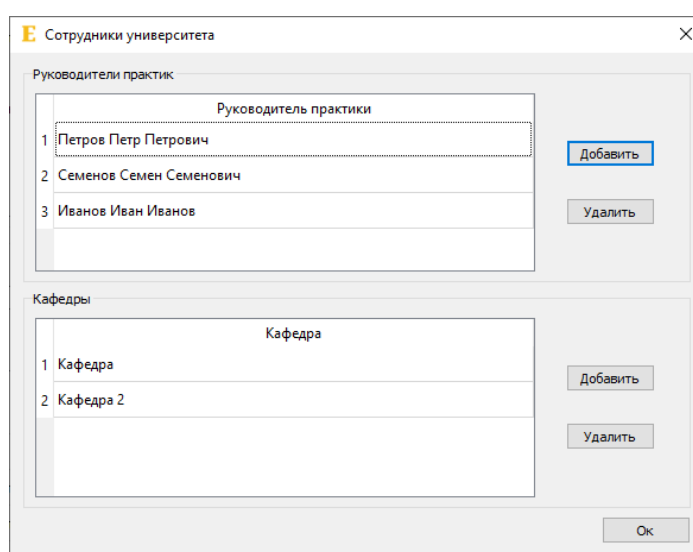


Рисунок 37 – Окно «Сотрудники университета»

В данном окне можно добавить руководителя практики, нажав на кнопку «Добавить» и заполнив форму, аналогичную рисунку 19, удалить руководителя практики, выбрав руководителя из таблицы и нажать на «Удалить», просмотреть или изменить руководителя практики, сделав двойное нажатие на руководителя практики из таблицы. Также в данном окне можно добавить кафедру, нажав на кнопку «Добавить, удалить кафедру, выбрав кафедру из таблицы и нажать на «Удалить», просмотреть или изменить кафедру, сделав двойное нажатие на кафедру из таблицы.

### 3.1.9 Студенты

При нажатии из главного окна на пункт меню «Студенты» откроется окно «Студенты» (рисунок 38), в котором отображаются все студенты и группы.

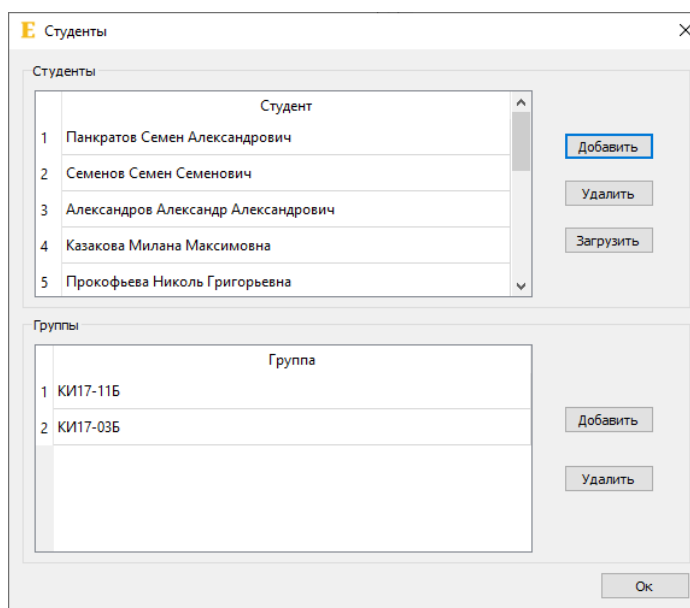


Рисунок 38 – Окно «Студенты»

В данном окне можно добавить студента, нажав на кнопку «Добавить» и заполнив форму, аналогичную рисунку 19, удалить студента, выбрав студента из таблицы и нажать на «Удалить», просмотреть или изменить студента, сделав двойное нажатие на студента из таблицы. Также в данном окне можно добавить группу нажав на кнопку «Добавить, удалить группу, выбрав группу из таблицы и нажать на «Удалить», просмотреть или изменить группу, сделав двойное нажатие на группу из таблицы.

В системе присутствует возможность загрузить студентов из файла, формата CSV. Для этого в окне «Студенты» необходимо нажать на «Загрузить», в появившемся окне необходимо выбрать формат CSV файла и сам файл.

### 3.2 Инструкция для администратора

Для входа в панель администратора в окне (рисунок 16) необходимо нажать на кнопку «Администратор» и ввести пароль для пользователя «postgres». После успешного входа появиться окно «Администратор» (рисунок 39).

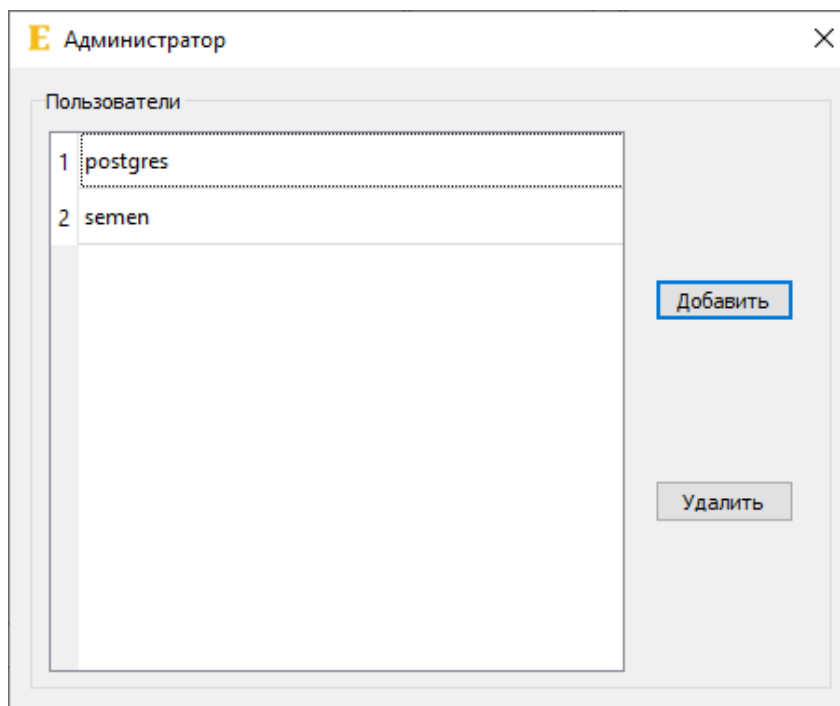


Рисунок 39 – Окно «Администратор»

Для добавления нового пользователя необходимо нажать на «Добавить» и заполнить форму (Рисунок 40).

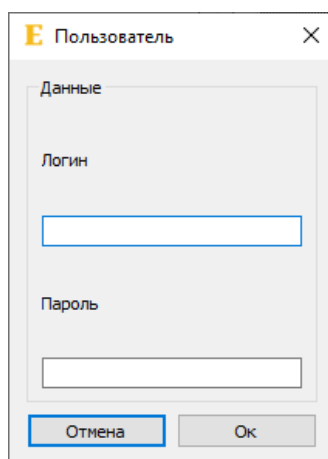
The image shows a standard Windows-style dialog box with the title bar 'Пользователь' (User) and a close button 'X'. The main area is titled 'Данные' (Data) and contains two text input fields. The first field is labeled 'Логин' (Login) and the second is labeled 'Пароль' (Password). At the bottom of the dialog, there are two buttons: 'Отмена' (Cancel) on the left and 'Ок' (OK) on the right.

Рисунок 40 – Окно «Пользователь»

Для удаления пользователя необходимо выбрать пользователя из таблицы и нажать на «Удалить».

Для изменения пароля пользователя необходимо сделать двойное нажатие на пользователя из таблицы и ввести новый пароль.

Для изменения адреса базы данных PostgreSQL нужно в файле /conf.ini вместо localhost указать адрес базы данных.

### 3.3 Инструкция для разработчика

Для сбора исходного кода необходима библиотека Qt версии 5.12.2, версия 6.0 не поддерживается. В качестве IDE для разработки и сборки можно использовать Qt Creator. Загрузить Qt вместе с Qt Creator можно с официального сайта Qt [6]. В качестве компилятора под Windows лучше выбрать MSVC 2019, под Linux GCC.

Исходники программы можно найти в репозитории GitHub [12]. Для работы программы также необходима библиотека QxOrm [11].

После скачивания библиотеки в корне проекта необходимо создать каталог QxOrm и переместить исходные файлы библиотеки в этот каталог. Далее нужно собрать саму библиотеку. Для этого можно воспользоваться файлом QxOrm.pro, запустив его в IDE Qt Creator. Собрать библиотеку в конфигурациях «Debug» и «Release». Для проверки можно проверить наличие файлов QxOrm и QxOrmd в каталоге lib.

Далее нужно перейти в каталог entityVKR и собрать библиотеку классов для работы с QxOrm. Для этого можно воспользоваться файлом department.gen.pro, запустив его в IDE Qt Creator. Собрать библиотеку в конфигурациях «Debug» и «Release». Для проверки можно проверить файлы department и departmentd в каталоге bin.

После этих действий необходимо собрать саму программу, воспользовавшись файлом clientVKR.pro, после этих действий программа должна запуститься.

Описание каталогов проекта:

- /entityVKR – библиотека классов для работы с QxOrm;
- /entityVKR/include – заголовочные файлы;
- /entityVKR/src – файлы реализации;
- /icons – каталог с расположением иконки;
- /include/view – заголовочные файлы классов графического интерфейса;
- /include/model – заголовочные файлы классов для работы с данными;
- /src/view – файлы реализации классов графического интерфейса;
- /src/model – файлы реализации классов для работы с данными;
- /ui – каталог с формами окон.

Для изменения или добавления функциональности над данными рекомендуется изменять соответствующую модель /include/model и /src/model.

Для изменения полей таблиц необходимо изменять файлы библиотеки классов для работы с QxOrm. Полями таблицы являются поля классов. Добавления или изменения полей можно осуществить аналогично с уже

существующими полями либо взять примеры из документации к библиотеке QxOrm.

Для добавления новых таблиц необходимо добавить новый заголовочный файл класса в `/entityVKR/include` и соответствующий файл реализации в `/entityVKR/src`. Оформить их можно по аналогии с уже существующими классами либо взять пример из документации к библиотеке QxOrm. Также для успешной сборки библиотеки добавленные файлы необходимо внести в `department.gen.pri`.

Новые формы окон необходимо добавлять в каталог `/ui`.

## ЗАКЛЮЧЕНИЕ

В результате работы была спроектирована и реализована система для учёта связей с работодателями.

Система обладает следующими функциями:

- ввод/вывод данных через графический интерфейс;
- хранение информации о практиках, работодателях, студентах, руководителях практик, мероприятиях, образовательных программах, дисциплинах;
- ввод данных студентов в машиночитаемом файловом формате CSV;
- поддержка многопользовательской работы;
- поиск практик, студентов, руководителей практик;
- создание отчетов о практиках и мероприятиях;
- возможность загружать тестовые задания от работодателей;
- возможность загружать отчеты студентов;

Возможные улучшения системы:

- синхронизация с LMS Moodle;
- возможность ввода прочих данных, помимо данных студентов, в машиночитаемом файловом формате;
- добавление возможностей визуализации и анализа.

Последняя версия клиентской части системы размещена в git-репозитории [11].



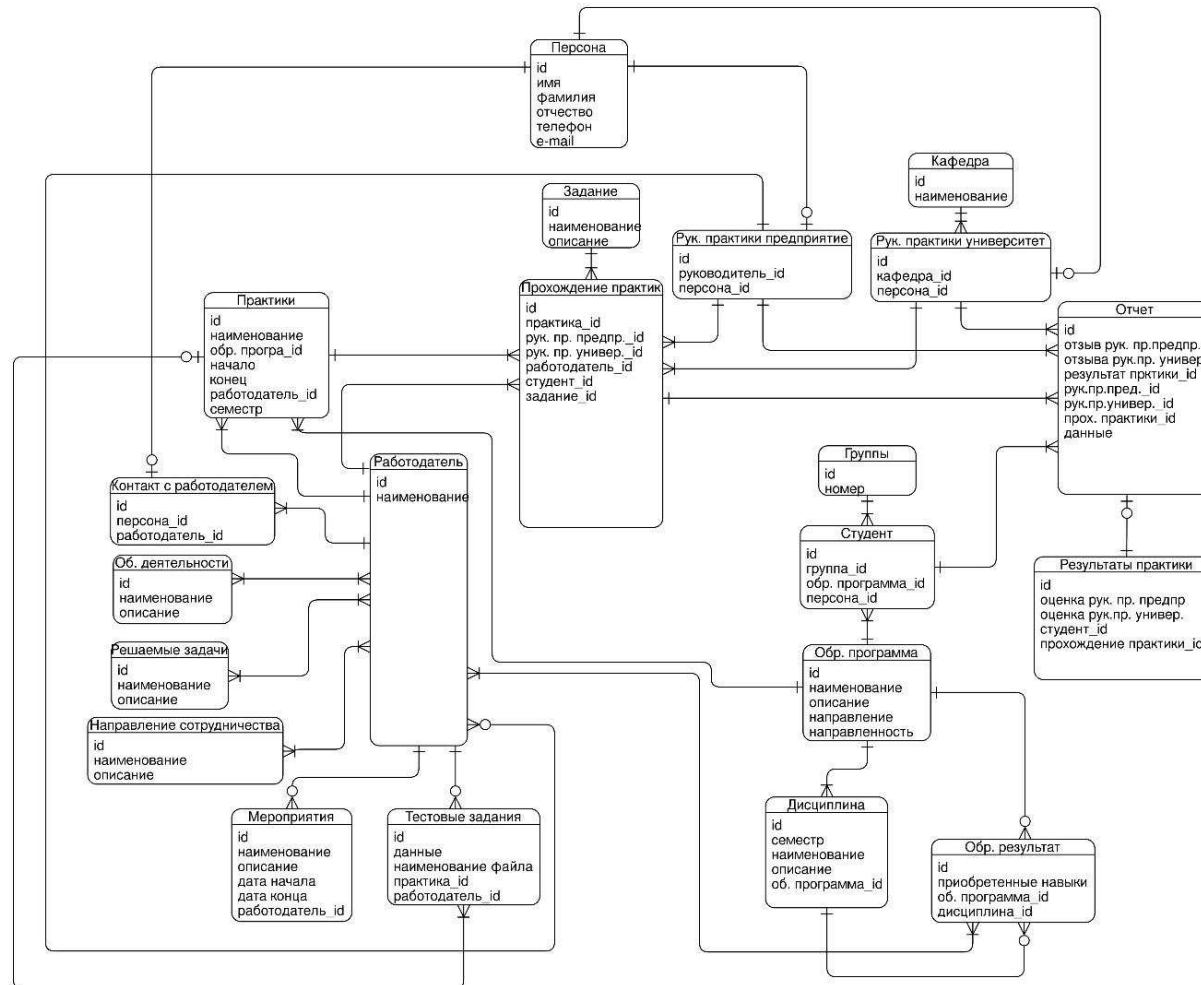
## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Кондрашина, Е. А. Информационная система «Вуз — работодатель» [Электронный ресурс] / Е. А. Кондрашина, А. В Кравченко. – Режим доступа: [http://www.appliedinformatics.ru/r/free/article/index.php?article\\_id 4=1248](http://www.appliedinformatics.ru/r/free/article/index.php?article_id 4=1248) (дата обращения: 22.11.20).
2. Рыжикова, Е. Г. Проектирование информационной системы для подбора персонала [Электронный ресурс] / Е. Г. Рыжикова, И. И. Теремкова, Д. А. Каршков. – Режим доступа: <https://cyberleninka.ru/article/n/proektirovanie-informatsionnoy-sistemy-dlya-podbora-personala/viewer> (дата обращения: 22.11.20).
3. Данилаев, Д. П. Система информационного взаимодействия вузов, работодателей и студентов [Электронный ресурс] / Д. П. Данилаев, Л. Ю. Емалетдинова. – Режим доступа: <https://cyberleninka.ru/article/n/sistema-informatsionnogo-vzaimodeystviya-vuzov-rabotodateley-i-studentov/viewer> (дата обращения: 22.11.20).
4. Java | Oracle [Электронный ресурс]. – Режим доступа: <https://www.java.com/ru/> (дата обращения: 29.11.20).
5. Парадигмы программирования в C++ | ITandLife.ru [Электронный ресурс]. – Режим доступа: <https://itandlife.ru/programming/cpp/paradigmy-programmirovaniya-v-c/> (дата обращения: 06.06.21).
6. Qt | Cross-platform software development for embedded & desktop [Электронный ресурс]. – Режим доступа: <https://www.qt.io> (дата обращения: 09.01.21).
7. Welcome to Python.org [Электронный ресурс]. – Режим доступа: <https://www.python.org> (дата обращения: 09.01.21).
8. SQLite - Краткое руководство - CoderLessons.com [Электронный ресурс]. – Режим доступа: <https://coderlessons.com/tutorials/bazy-dannykh/vyuchit-sqlite/sqlite-kratkoe-rukovodstvo> (дата обращения: 22.12.20).
9. MySQL [Электронный ресурс]. – Режим доступа: <https://www.mysql.com> (дата обращения: 29.11.20).

10. PostgreSQL: The world's most advanced open source database [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org> (дата обращения: 09.01.21).
11. GitHub - QxOrm/QxOrm: QxOrm library - C++ Qt ORM (Object Relational Mapping) and ODM (Object Document Mapper) library - Official repository [Электронный ресурс]. – Режим доступа: <https://github.com/QxOrm/QxOrm> (дата обращения 24.05.2021).
12. GitHub - 7enSmile/clientVKR [Электронный ресурс]. – Режим доступа: <https://github.com/7enSmile/clientVKR> (дата обращения 24.05.2021).

# ПРИЛОЖЕНИЕ А (обязательное)

## Логическая модель базы данных




Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

  
О. В. Непомнящий

подпись


« 08 » 06 2021 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 – «Информатика и вычислительная техника»

Система для учета связей с работодателями


Руководитель

  
07.06.2021  
подпись, дата

ст. преподаватель


К. В. Пушкарев

Выпускник

  
07.06.2021  
подпись, дата

С. А. Панкратов

Нормоконтролер

  
07.06.2021  
подпись, дата

К. В. Пушкарев

Красноярск 2021