Казанский (Приволжский) федеральный университет

Институт вычислительной математики и информационных технологий

Кафедра системного анализа и информационных технологий

**Отчет по лабораторной работе №1.**

**«Введение в объектно-ориентированный анализ и проектирование.**

**Язык UML».**

Выполнил:

Плотников А. И.

группа 09-913

Казань - 2023.

**Описание проекта**

В качестве проекта для разработки диаграмм на языке UML использовалось приложение по скрапингу вакансий.

Были разработаны следующие диаграммы:

1. Диаграмма прецедентов (вариантов использования) (рис. 1)
2. Диаграмма последовательностей (рис. 2)
3. Диаграмма деятельности (рис. 3)
4. Диаграмма таблиц базы данных (рис. 4)
5. Диаграмма деятельности

**Диаграмма прецедентов**

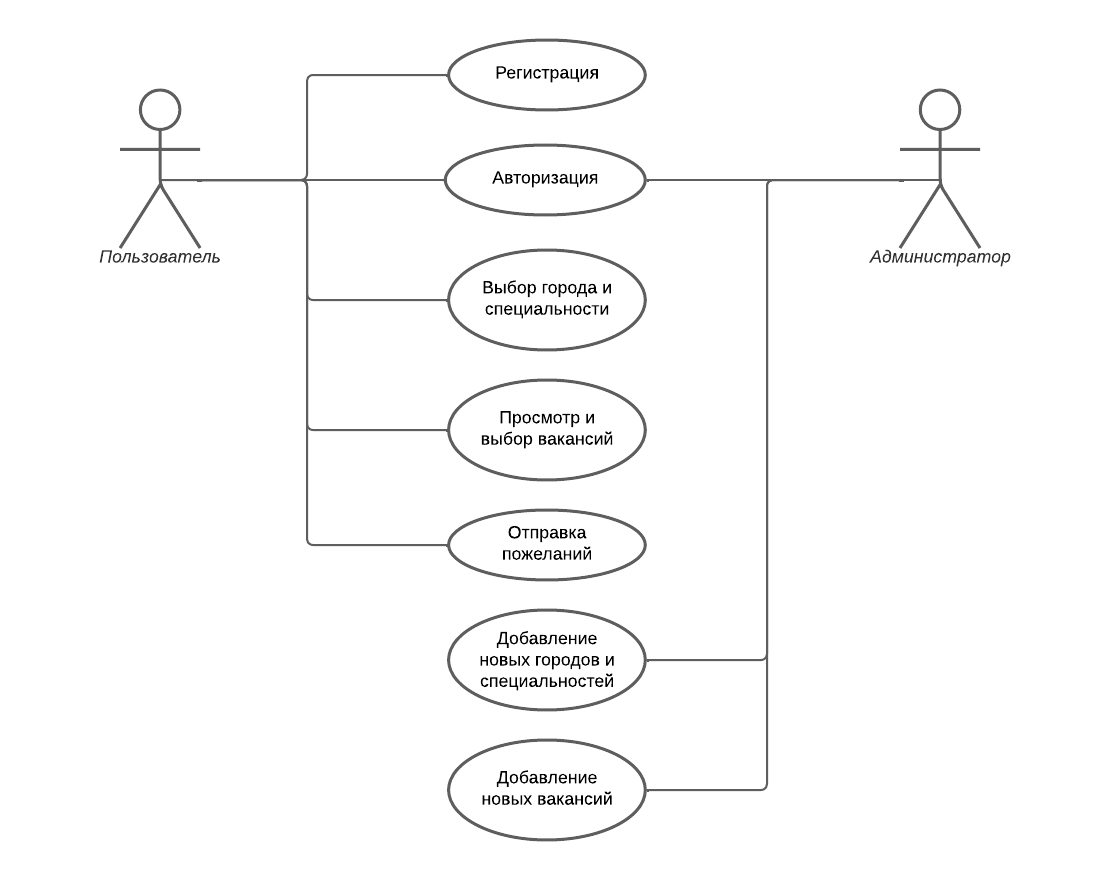


Рисунок 1. Диаграмма прецедентов

Рассмотрим описание всех прецедентов, описанных на данной диаграмме.

### **Регистрация**

Краткое описание:

Данный прецедент описывает процесс добавления нового пользователя в систему.

Основной поток:

1. Пользователь может открыть окно регистрации, нажав на кнопку «Регистрация»
2. Пользователь заполняет свой адрес электронной почты, пароль, подтверждает его и нажимает на кнопку «Зарегистрироваться»
3. Система сверяет пароль и подтверждённый пароль
4. При успешной регистрации открывается окно с кнопкой, переадресовывающей на главную страницу сервиса.

Альтернативные потоки:

## При неправильной форме ввода адреса электронной почты выводится сообщение об этом

* + 1. При незаполненных полях выводится сообщение о необходимости их заполнения
    2. При вводе уже существующего в системе адреса электронной почте выводится сообщение об этом

### **Авторизация**

я

Основной поток:

1. Пользователь может открыть окно авторизации, нажав на кнопку «Войти»
2. Пользователь заполняет адрес электронной почты и пароль.
3. Пользователь нажимает на кнопку «Войти»
4. После успешной авторизации происходит перенаправление на главную страницу сервиса.

Альтернативные потоки:

1. При незаполненных полях выводится сообщение о необходимости их заполнения
2. При вводе не зарегистрированного в системе адреса электронной почты выводится сообщение о том, что такой аккаунт не найден
3. При вводе неверного пароля выводится сообщение об этом

### **Выбор города и специальности**

Краткое описание:

Пользователь выбирает интересующие город и специальность для отображения подходящих по этим параметрам вакансий.

Основной поток:

1. Пользователь в окне выбирает один из существующих городов
2. Пользователь в окне выбирает одну из существующих специальностей
3. Пользователь нажимает на кнопку «Найти»
4. Система выводит список вакансий по выбранным параметрам

Альтернативный поток:

1. При незаполненных полях выводится сообщение о том, что по такому запросу ничего не найдено
2. При незаполненном одном из полей выводится список всех вакансий с выбранного города или с выбранной специальности

### **Просмотр и выбор вакансий**

# Краткое описание:

# Пользователь просматривает список вакансий и выбирает заинтересовавшие его предложения.

Основной поток:

1. Пользователь кликает на название заинтересовавшей его вакансии
2. Открывается страница сайта, с которого была получена информация об этой вакансии

### **Отправка пожеланий**

# Краткое описание:

Пользователь может не обнаружить в предлагаемых списках интересующие его город и специальность, тогда он может отправить пожелание с указанием, названия каких города и специальности он бы хотел добавить в этот сервис.

Основной поток:

1. Пользователь заходит в свой профиль, нажав на соответствующий пункт меню
2. Пользователь нажимает на кнопку «Показать форму», после чего открывается форма
3. Пользователь заполняет необходимые поля: город, специальность и адрес электронной почты, куда будет отправлено письмо после добавления этих параметров
4. Пользователь нажимает на кнопку «Отправить»
5. Сообщение с введёнными данными сохраняется в базе данных системы

Альтернативный поток:

1. При незаполненных полях выводится сообщение о необходимости их заполнения

### **Добавление новых городов и специальностей**

Краткое описание:

Администратор может добавить новые города и специальности.

Основной поток:

1. Администратор заходит в панель администратора
2. Администратор добавляет новый город или специальность либо изменяет каким-либо образом уже существующие
3. Администратор нажимает на кнопку «Сохранить» и в базу данных сохраняются новые данные

### **Добавление новых вакансий**

1. Администратор может либо добавить вручную новую вакансию с указанием города и специальности для неё, либо выбрать город и специальность и запустить скрапинг с сайтов, тем самым добавив оттуда все вакансии с выбранными городом и специальностью. Так же он может внести какие-то изменения в уже добавленные вакансии.
2. Администратор нажимает на кнопку «Сохранить» и изменения сохраняются в базу данных

**Диаграмма последовательностей.**

**Диаграмма последовательностей** служит для представления взаимодействия элементов модели в форме последовательности сообщений и соответствующих событий на линиях жизни объектов. Обычно диаграмма последовательности строится для каждого из вариантов использования, конкретизируя его выполнение

С помощью данной диаграммы описан процессы, которые протекают в системе поиска вакансий.

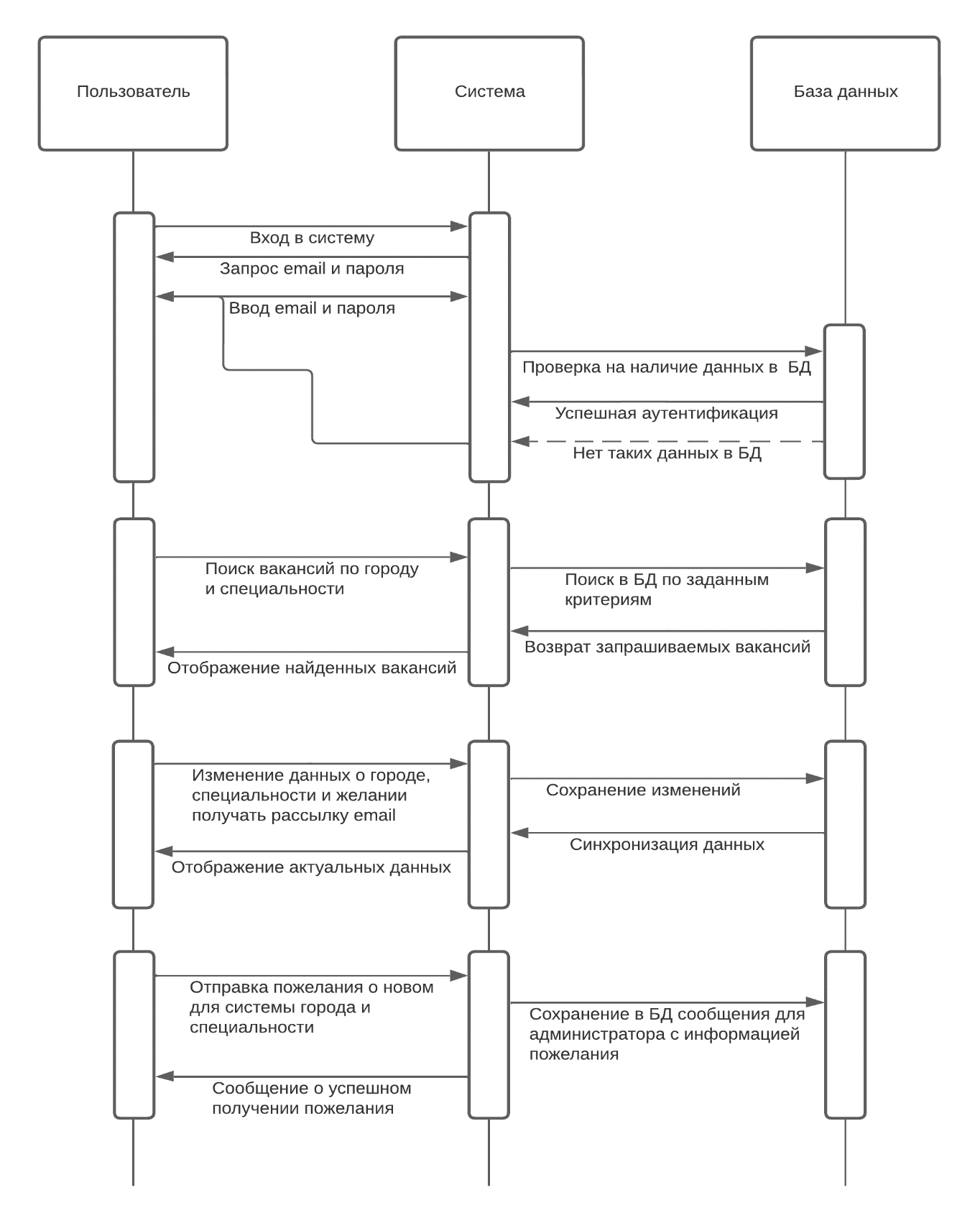
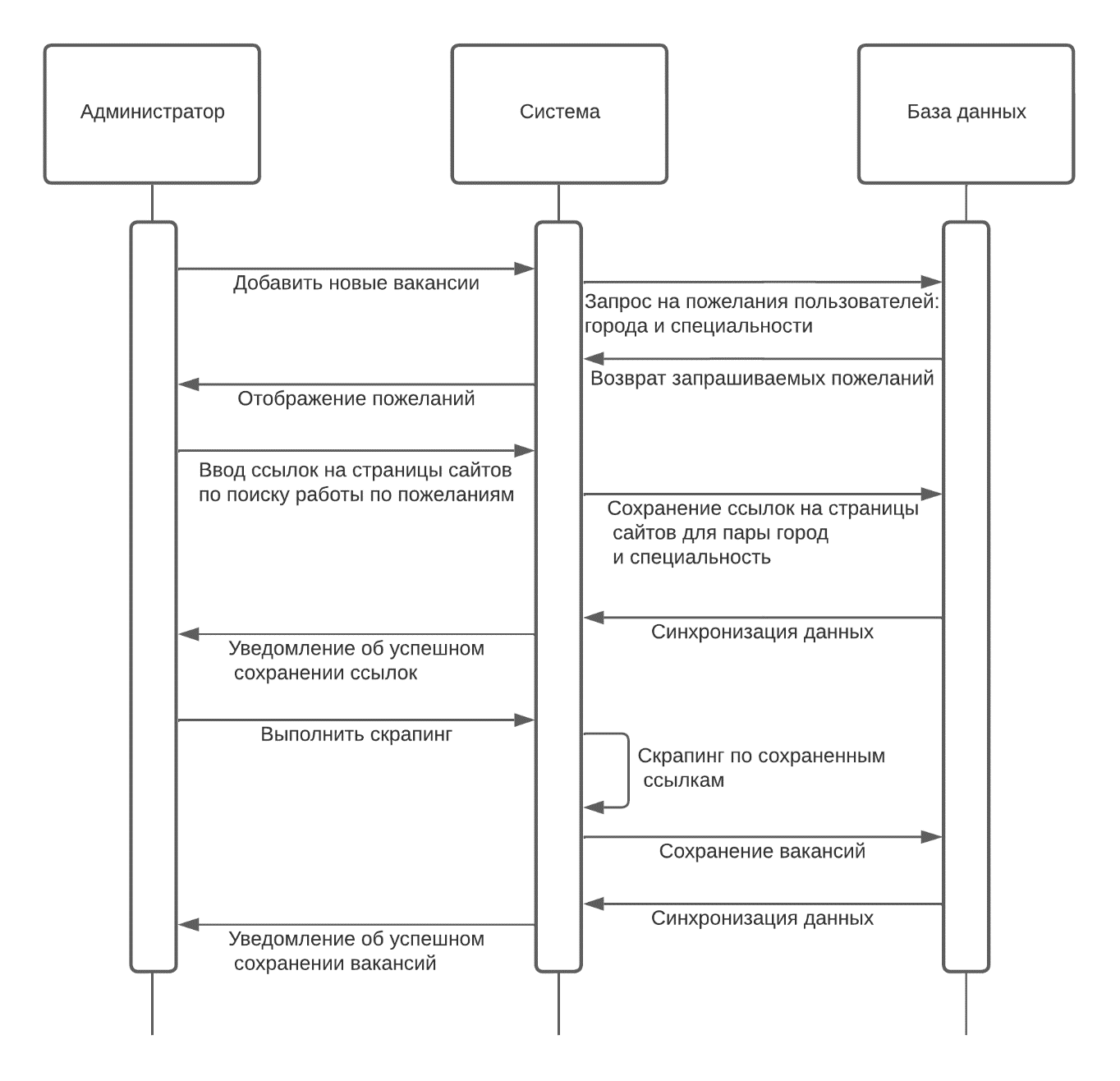


Рисунок 2. Диаграмма последовательностей пользователя

**** Рисунок 3. Диаграмма последовательностей администратора

**Диаграмма связей таблиц базы данных**

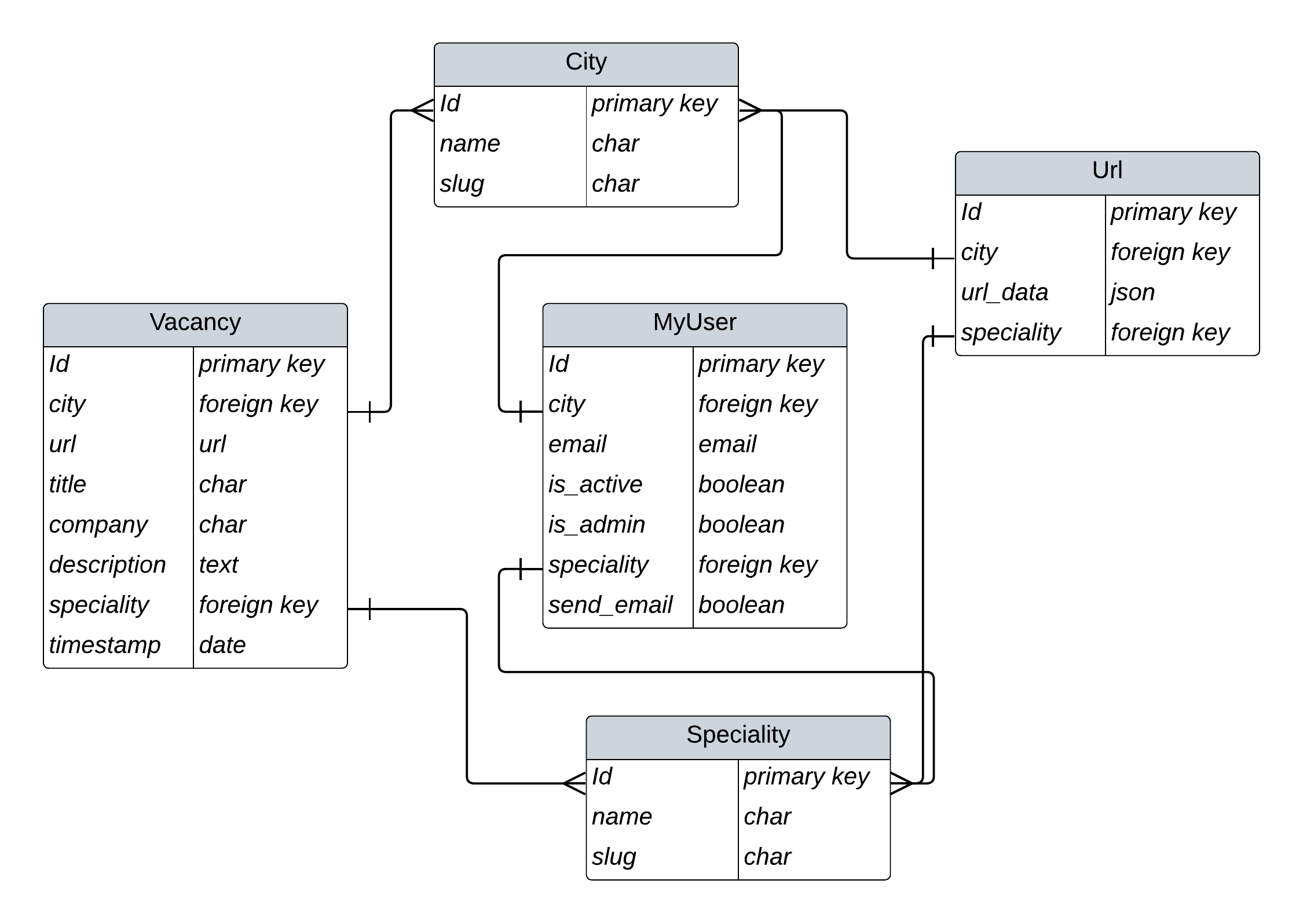
****

Рисунок 4. Схема базы данных

Описание полей таблиц:

Таблица «City»:

* Id – индивидуальный номер города
* name – название города
* slug – уникальный идентификатор, представляющий из себя название города в нижнем регистре

Таблица «Speciality»:

* Id – индивидуальный номер специальности
* name – название специальности
* slug – уникальный идентификатор, представляющий из себя название специальности в нижнем регистре

Таблица «Vacancy»:

* Id – индивидуальный номер вакансии
* url – ссылка на вакансию в сайте по поиску работы
* title – название вакансии
* company – компания, разместившая данную вакансию
* description – описание вакансии
* city – внешний ключ, связанный с таблицей «City»
* speciality – внешний ключ, связанный с таблицей «Speciality»
* timestamp – время, когда была опубликована данная вакансия

Таблица «Url»:

* Id – индивидуальный номер ссылки на вакансию
* city – внешний ключ, связанный с таблицей «City»
* speciality – внешний ключ, связанный с таблицей «Speciality»
* url\_data – набор ссылок на страницы сайтов по поиску работы для уникальных наборов город и специальность

Таблица «MyUser»:

* Id – индивидуальный номер пользователя
* email – email адрес пользователя
* is\_active – показатель, является ли этот пользователь активным
* is\_admin – показатель, является ли этот пользователь админом
* city – внешний ключ, связанный с таблицей «City»
* speciality – внешний ключ, связанный с таблицей «Speciality»
* send\_email – показатель, желает ли пользователь получать рассылку о вакансиях на почту

**Диаграмма деятельности**

Следующий вид диаграмм — это диаграмма деятельностей, описывающая действия персонала в системе

Диаграмма деятельности вида хорошо отражает:

* последовательность действий;
* события, инициирующие действия или являющиеся конечным результатом;
* условия расширения сценария;

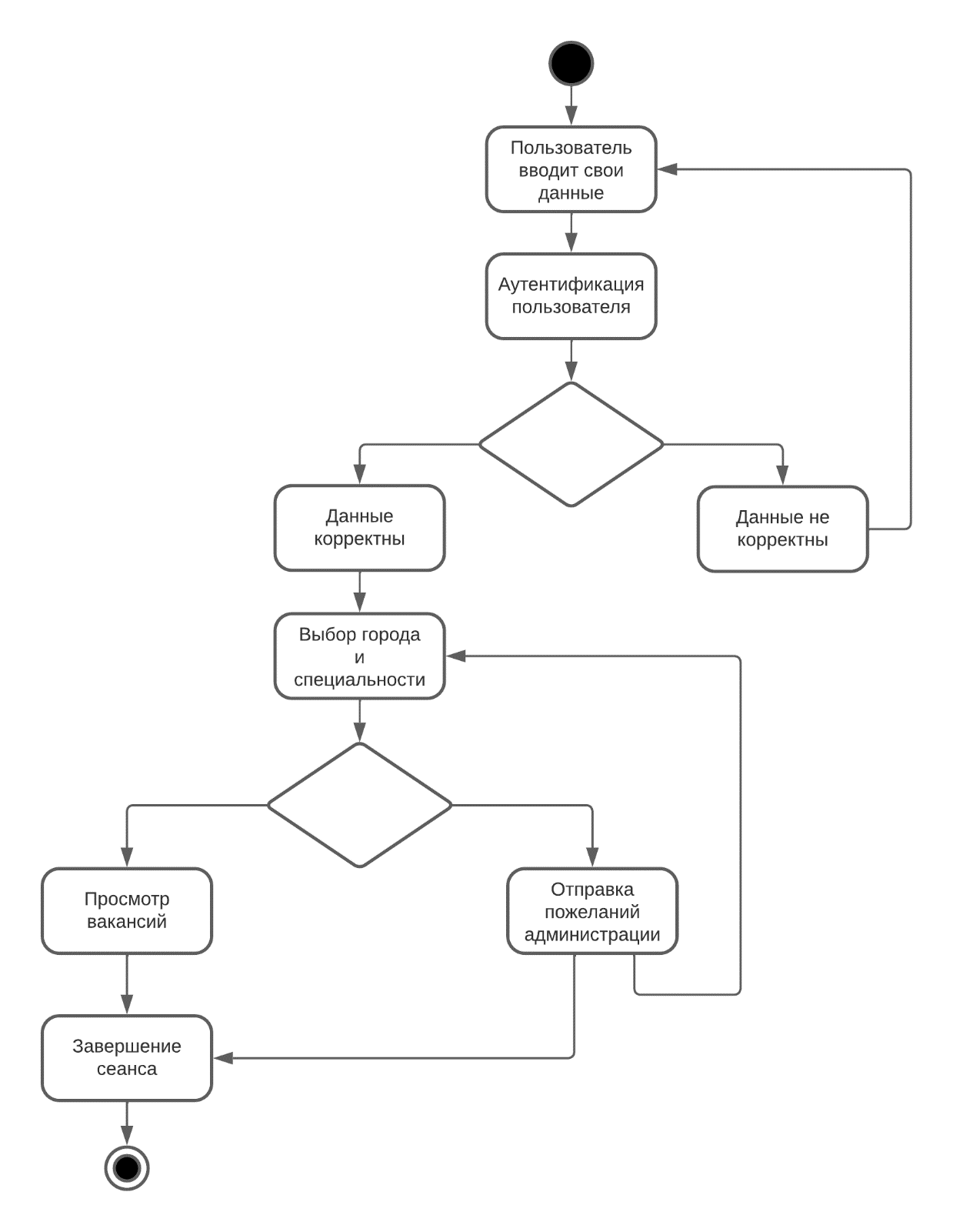


Рис. 5 Диаграмма деятельности

Ссылка на GitHub проекта: <https://github.com/Rarn1k/django_job>