Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»

Образовательная программа «Веб-технологии»

ПРАКТИЧЕСКОЕ ЗАДАНИЕ ПО ДИСЦИПЛИНЕ

 «Программная инженерия»

на тему:

 «Лабораторная работа №4.

Объектно-ориентированный подход к моделированию бизнес процессов.

Модель классов и модель прецедентов.»

**Выполнил:**

Студент группы 181-321

                                      Гусев Ф.А.

Москва 2020

1. **Назначение программы и цели ее создания:** Данное программное обеспечение разрабатывается для упрощения создания записей к врачам-ветеринарам, а также их учета и хранения в журнале посещений. К тому же, данный программный продукт позволит создавать и хранить записи о результатах осмотра питомца ветеринаром и выявленных диагнозах. Также у администратора будущей ИС должна присутствовать возможность формирования отчета на основе журнала записей.
2. **Перечень задач**:
   1. *Требования к задаче «Регистрация клиента»*

Администратор ветеринарной клиники может добавить нового клиента, указав его ФИО, контактные данные. Запись о новом клиенте сохраняется в базу данных.

* 1. *Требования к задаче «Добавление записи к ветеринару»*

Клиент может создать запись к ветеринару при помощи формы записи к ветеринару, указав свои ФИО и контактные данные. Также добавить запись о посещении ветеринара может администратор ветеринарной клиники по заявке клиента. Все данные вносятся в базу данных.

* 1. *Требование к задаче «Просмотр журнала записей»*

На экране просмотра записей к ветеринару администратору отображается вся информация о деталях записи – ФИО клиента, цель визита, время и дата. Другим пользователям системы отображается только время и дата записи, информирующая его о том, что на данное время запись уже есть. Все записи загружаются из базы данных.

* 1. *Требование к задаче «Изменение записи»*

Администратор ветеринарной клиники может изменять данные записи – ветеринара, цель посещения. Все данные изменяются в журнале посещений в базе данных.

* 1. *Требование к задаче «Перенос записи»*

Администратор ветеринарной клиники может перенести запись по просьбе клиента или по просьбе ветеринара при изменении графика приема. Запись о приеме переносится с одной даты на другую при наличии свободного времени и текущего графика приема ветеринара в журнале посещений в базе данных.

* 1. *Требование к задаче «Удаление записи»*

Администратор ветеринарной клиники может удалить запись по просьбе клиента. Запись о приеме удаляется из журнала посещений в базе данных.

* 1. *Требование к задаче «Добавление записи о результатах приема в медкарту»*

Ветеринар клиники может добавлять записи о результатах приема клиента в медицинскую карту, указав имя животного, цель приема, результаты осмотра и рекомендации по лечению. Все данные заносятся и хранятся в сущности медицинской карты в базе данных.

* 1. *Требование к задаче «Добавление питомца»*

Администратор и ветеринар могут добавлять нового питомца, указывая его имя и тип животного. Все данные хранятся в базе данных.

* 1. *Требование к задаче «Добавление ветеринара»*

При необходимости администратор может добавлять нового ветеринара, указывая его ФИО и специализацию.

* 1. *Требование к задаче «Добавление болезней в общий список болезней»*

При необходимости ветеринар или администратор ветеринарной клиники может добавлять наименование и характеристику болезни с общий список болезней, хранящийся в базе данных.

На основе вышеописанных требований составлена модель прецедентов (рис. для графического описания функциональных требований к программе. Выбранная среда моделирования: Microsoft Visio 2010.

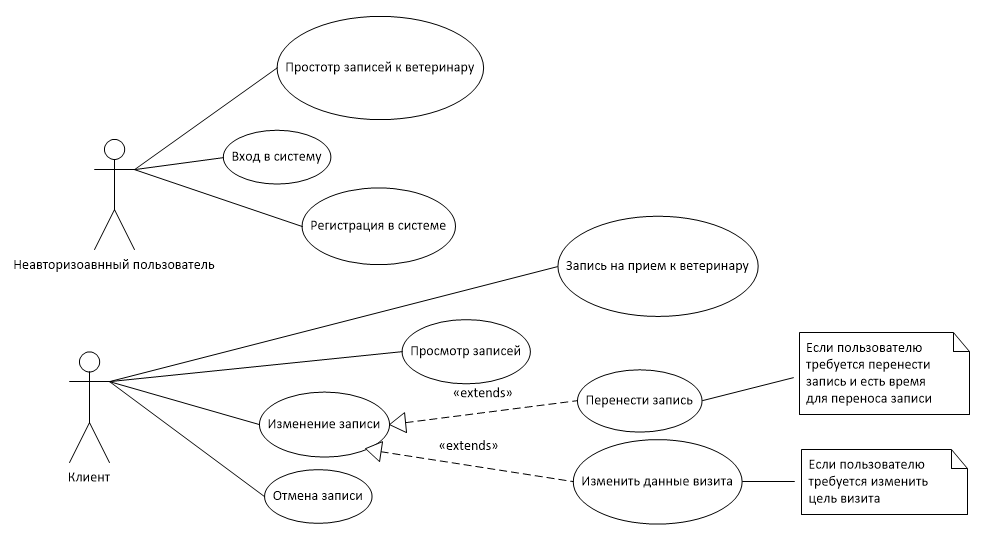


Рис. 1 – Диаграмма прецедентов (UML). Актеры «Клиент» и «Неавторизованный пользователь»

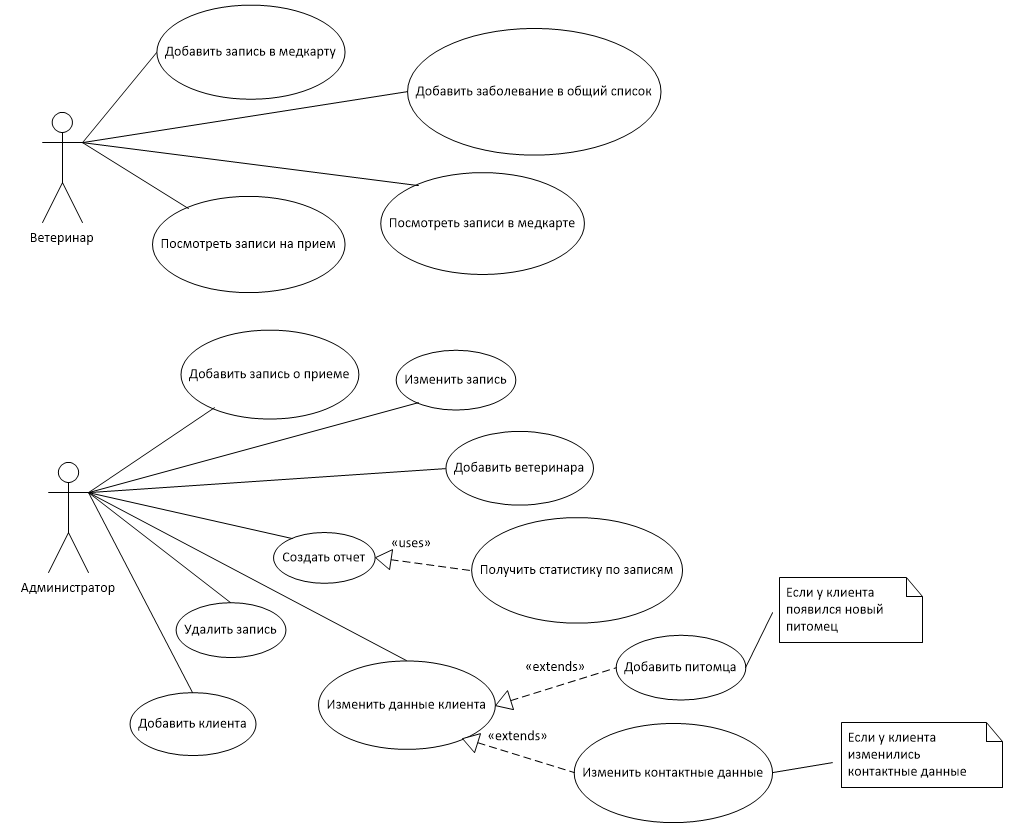


Рис. 2 - Диаграмма прецедентов (UML). Актеры «Ветеринар» и «Администратор»

В данной модели присутствуют следующие сущности:

1. Клиенты (clients) – содержит поля уникального ключа, фамилию, имя клиента, а также его контактные данные.
2. Ветеринары (veterinarians) – хранит id, ФИО и специализацию врача-ветеринара.
3. Питомцы (pets) – содержит уникальный номер, а также имя питомца и тип животного.
4. Болезни (diseases) – хранит в себе общий список всех болезней, который применяется при постановке диагноза животному. Содержит поля id, название болезни и описание.
5. Медицинская карта (medical\_record) – содержит записи обо всех результатах приема ветеринара. Имеет уникальное поле записи, результат осмотра, а также вторичные ключи к списку болезней, питомцам и номеру записи в журнале записей из соответствующих таблиц.
6. Журнал записей (appointments) – содержит уникальный номер записи, дату и время записи, а также вторичные ключи к клиенту и ветеринару из соответствующих таблиц.

На основе выделенных сущностей создана модель сущность-связь (ERD) (рис. 3). Выбранная среда моделирования: Microsoft Visio 2010.

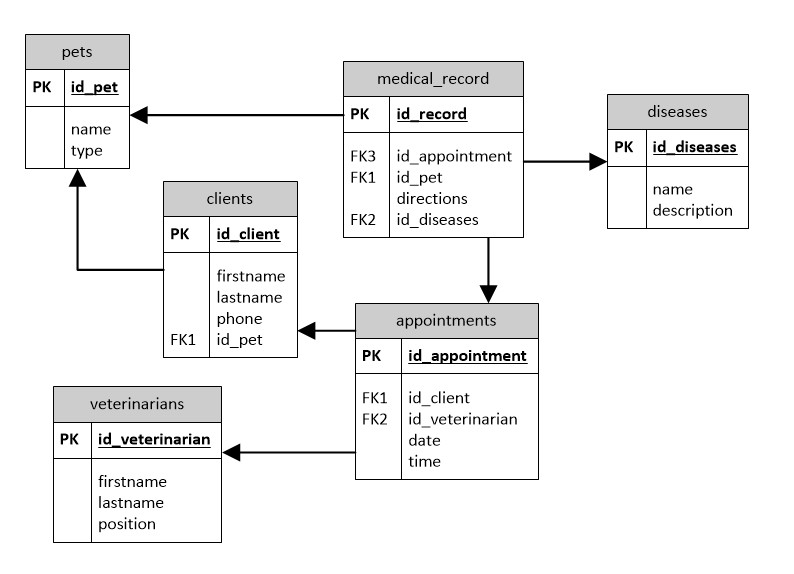


Рис. 3 – Модель сущность-связь (ERD)

Таким образом, на основе требований к разрабатываемому программному обеспечению было создано их графическое представление в виде модели прецедентов (UML) и модели сущность-связь (ERD).