

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

Краснова О. В.

Москва 2020

1. Определение структуры программы и состав функциональных задач.

Перечень функциональных требований к программной реализации информационной системе для парикмахерской

1. Требования к задаче “Авторизация клиента”

Введённые данные сравниваются с привязанной базой данных. Когда введены некорректные данные (пустые поля, неправильный логин или пароль) пользователю демонстрируется сообщение с типом ошибки. Когда введены корректные данные и они существуют в базе данных, пользователя переводит на страницу “личный кабинет”. На странице отображается имя и роль пользователя.

2. Требования к задаче “Выход из системы”

При нажатии кнопки “Выйти” в боковом меню пользователь завершает сессию для своей учетной записи и возвращается на главную страницу.

3. Требования к задаче “Запись к мастеру”

Авторизованный клиент может записаться к мастеру через личный кабинет. Клиент выбирает дату, время, мастера, услугу, вводит данные о себе (ФИО, email, телефон). После этого выводится модальное окно “Вы успешно записаны. Ваш номер записи: *номер записи*”.

Запись вносится в БД, мастер становится недоступным на записанное время, другой пользователь не может забронировать этого мастера на это же время.

4. Требования к задаче “Просмотр списка записей (клиент)”

В личном кабинете клиента должны отображаться записи, предназначенные только для него и они не должны отображаться другим пользователям информационной системы.

5. Требования к задаче “Отмена записи”

Клиент может отменить запись в личном кабинете. Запись удаляется из базы данных.

6. Требования к задаче “Просмотр списка записей (мастер)”

В личном кабинете мастера должны отображаться записи, предназначенные только для него и они не должны отображаться другим клиентам.

7. Требования к задаче “Просмотр списка записей (администратор)”

В личном кабинете администратора должны отображаться все записи, отсортированные по мастерам.

8. Требования к задаче “Изменение статуса записи”

Мастер может зайти в личный кабинет, просмотреть записи, относящиеся непосредственно к нему, присвоить статус “выполнено” к выполненной записи.

9. Требования к задаче “Регистрация нового мастера”

Администратор может добавить нового мастера, указав его ФИО, роль в информационной системе, логин и пароль. Запись о новом мастере сохраняется в базе данных.

10. Требования к задаче “Редактирование учетных данных мастера”

Администратор в личном кабинете может получить список всех мастеров и редактировать информацию о них. Информация обновляется в базе данных.

11. Требования к задаче “Удаление мастера”

Администратор в личном кабинете может получить список всех мастеров и удалить мастера. Информация обновляется в базе данных.

12. Требования к задаче “Редактирование времени для записи”

Администратор в личном кабинете может получить список, содержащий дни и доступное время, и отредактировать время. Информация обновляется в базе данных.

13. Требования к задаче “Редактирование дат для записей”

Администратор в личном кабинете может получить список, содержащий дни, и добавить или удалить дни. Информация обновляется в базе данных.

14. Требования к задаче “Печать списка записей”

Администратор или мастер в личном кабинете может нажать печать под списком записей и получить pdf-файл с записями.

15. Требования к задаче “Добавление услуги”

Администратор в личном кабинете может просмотреть список выполняемых услуг мастером и добавить новую услугу. Информация обновляется в базе данных.

2. Разработка модели классов и модели прецедентов для графического описания функциональных требований к программе.

Для разработки моделей был использован сервис draw.io
(<https://app.diagrams.net>)

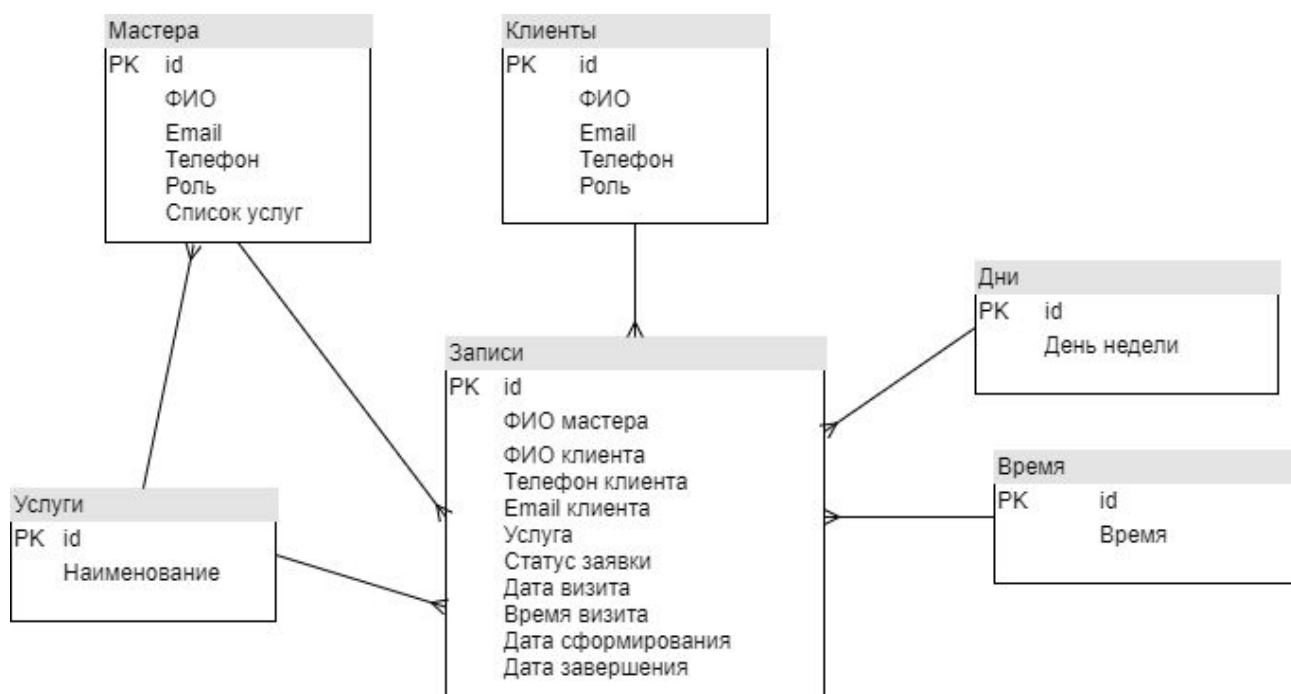


Рисунок 1. Модель классов (ERD диаграмма)



Рисунок 2. Модель прецедентов