

## Оглавление

Введение.....	2
Цель работы.....	3
Структура сайта.....	4
Требования.....	4
Диаграммы прецедентов.....	4
Поток событий для прецедента.....	6
Диаграммы .....	8
Диаграммы последовательности .....	8
Динамические аспекты системы.....	8
ПРОЕКТИРОВАНИЕ БД.....	11
Сущности .....	11
Модели .....	14
Создание сайта с использованием технологии ASP.NET .....	16
Разметка сайта. Фильтрация данных.....	16
Реализация личного кабинета пользователя.....	20
Администрирование.....	25
Заключение .....	29
ПРИЛОЖЕНИЕ А .....	30
ПРИЛОЖЕНИЕ Б.....	31
ПРИЛОЖЕНИЕ В .....	32
ПРИЛОЖЕНИЕ Г .....	33
ПРИЛОЖЕНИЕ Д .....	35
ПРИЛОЖЕНИЕ Е.....	40
ПРИЛОЖЕНИЕ Ж .....	42
ПРИЛОЖЕНИЕ З .....	46
ПРИЛОЖЕНИЕ И .....	48

## **Введение**

В каждом городе России есть проблема с бездомными животными. Для того, чтобы не давать питомцам пострадать от улицы и одиночества, есть специальные выделенные места, где животные могут чувствовать себя в тепле и безопасности. Приюты всеми силами стараются рассказать о своих жителях, чтобы каждый кот или собака могли найти свой дом. Поэтому – сайт, где человек сможет ознакомиться с питомцем, или же просто поддержать их, необходим.

## **Цель работы**

Необходимо разработать модель сайта для питомника, в котором содержатся кошки. Данный сайт разрабатывается для того, чтобы люди могли узнать о всех проживающих в приюте и их историю, или пристроить туда нуждающегося в доме питомца. А также в случае чего, могли материально поддержать приют или поучаствовать в мероприятиях по волонтерству.

## Структура сайта

### Требования

Есть четыре категории пользователей: администрация приюта, зарегистрированные пользователи, гости и интернет-пользователь.

Гости могут просмотреть карточку каждого животного, в которой помимо основной информации, содержатся их фотографии, состояние здоровья и наличие прививок и в случае чего авторизоваться (или зарегистрироваться) и отправить заявку в администрацию приюта, чтобы забрать питомца себе. Также пользователь может просто поддерживать питомник материально или принять участие в волонтерстве. Кроме того, есть возможность сдать котика в приют.

Администрация приюта следит за обновлением информации, а также обрабатывает заявки от желающих приютить/сдать питомца и поучаствовать в мероприятиях по волонтерству.

### Диаграммы прецедентов

Общая диаграмма прецедентов на рисунке 1.

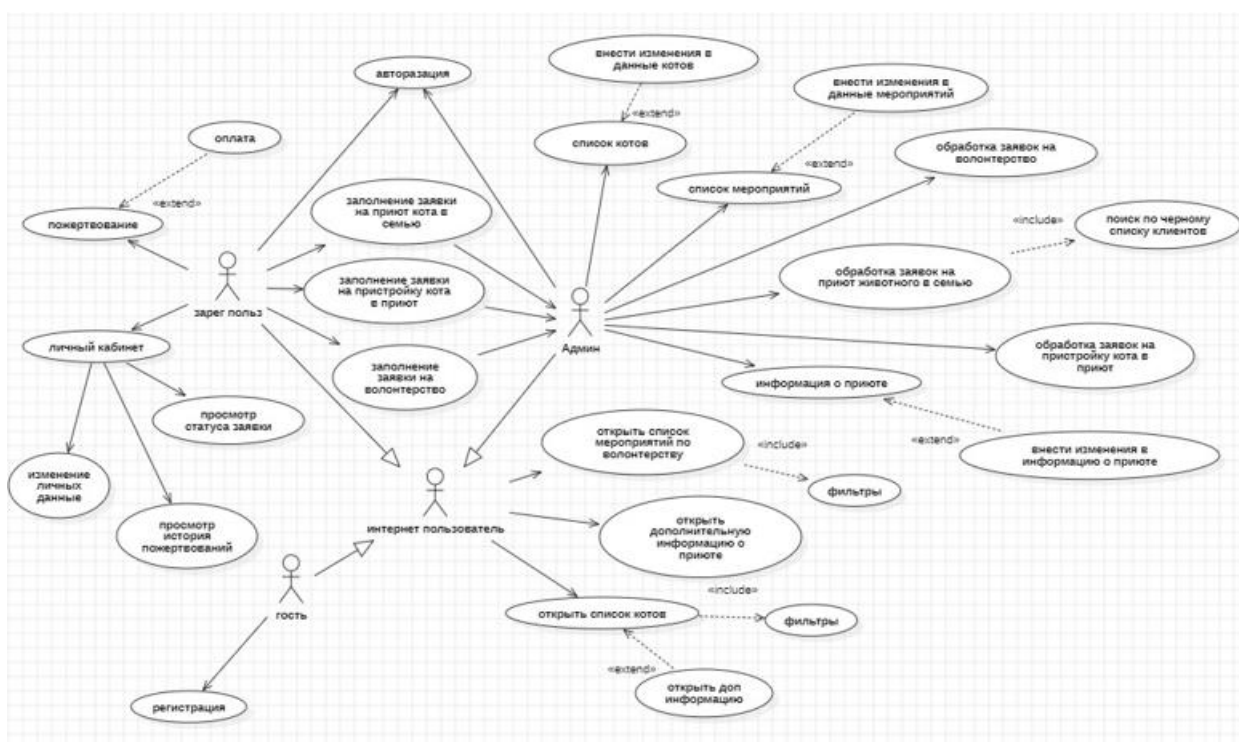


Рисунок 1 – Диаграмма прецедентов

Диаграммы прецедентов каждого актера на рисунках 2, 3 и 4.

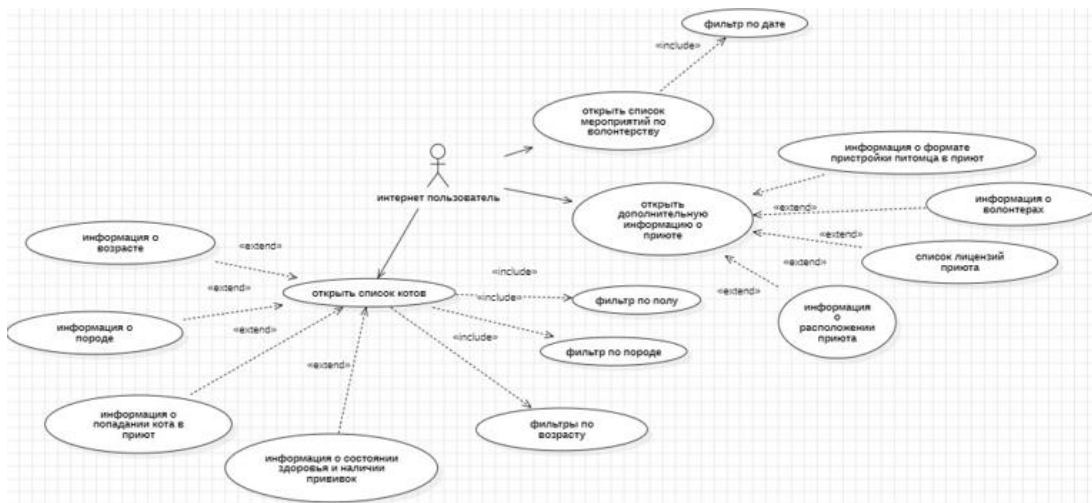


Рисунок 2 – Диаграмма прецедента Интернет-пользователя

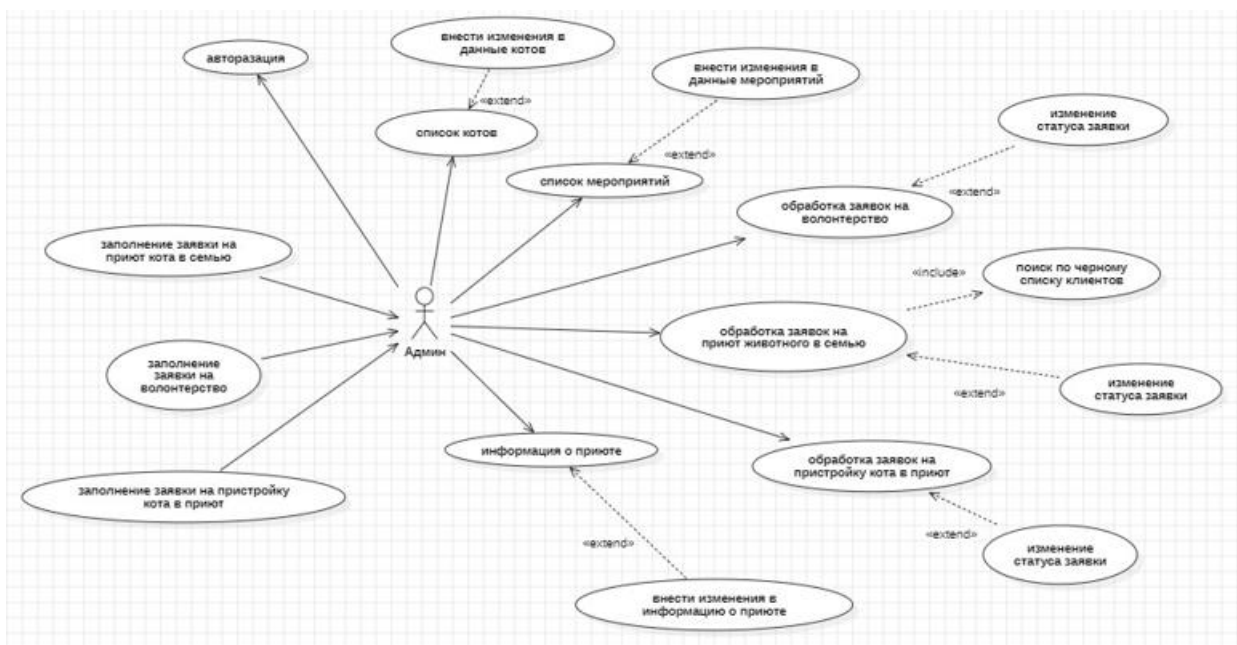


Рисунок 3 – Диаграмма прецедента Администратора

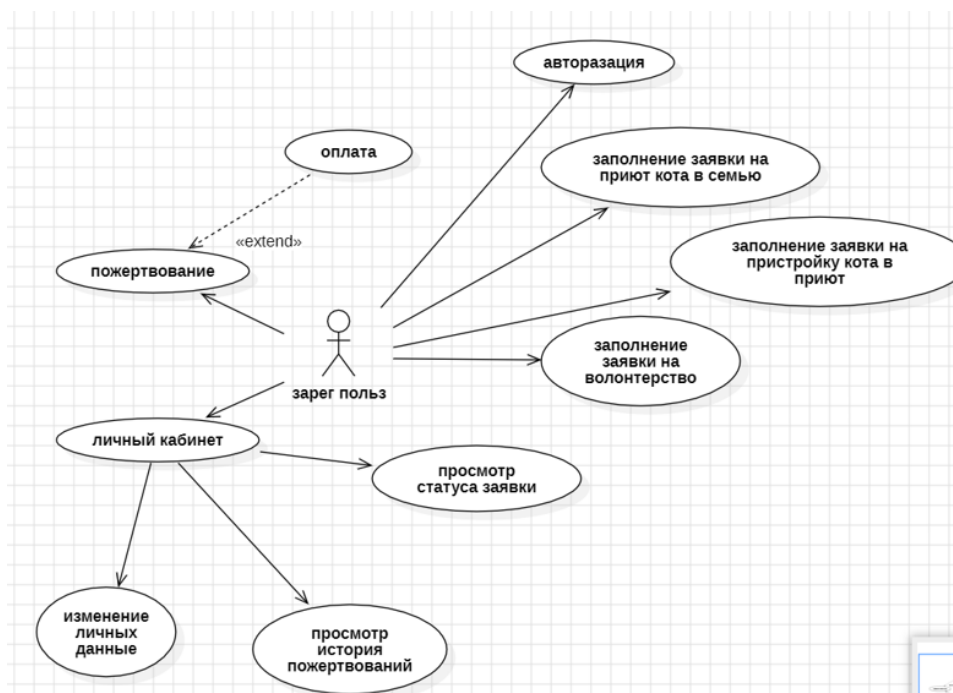


Рисунок 4 – Диаграмма прецедента зарегистрированного пользователя

## Поток событий для прецедента

### Поток событий

Краткое описание: прецедент “заполнения заявки на приют кота в семью” позволяет гостю отправить заявку на приют животного администрации приюта.

### Предусловие

Пользователь должен быть зарегистрирован в сети и находится в разделе “Дополнительной информации”

### Основной поток событий для прецедента “Заявка на кота”

1. Пользователь выбирает действие «Оформление заявки»
2. Необходимо ввести логин и пароль
3. Ввод логина и пароля
4. Если введенный пароль не верен, выполняется поток ошибок E1;
5. Если пароль верен система открывает форму на заполнение заявки;
6. Пользователь заполняет свои данные
7. Пользователь нажимает “отправить заявку”
8. Заявка отправляется администратору

9. Отправка сообщения пользователю об успешные отправки заявки
10. Пользователю предлагается перейти в личный кабинет для просмотра статуса заявки

Поток ошибок E1

Выход ошибка о неправильно введённом логине или пароле.

## Диаграммы

### Диаграммы последовательности

Диаграммы последовательности для прецедента “Заполнение заявки на приют кота в семью” представлена на рисунке 5.

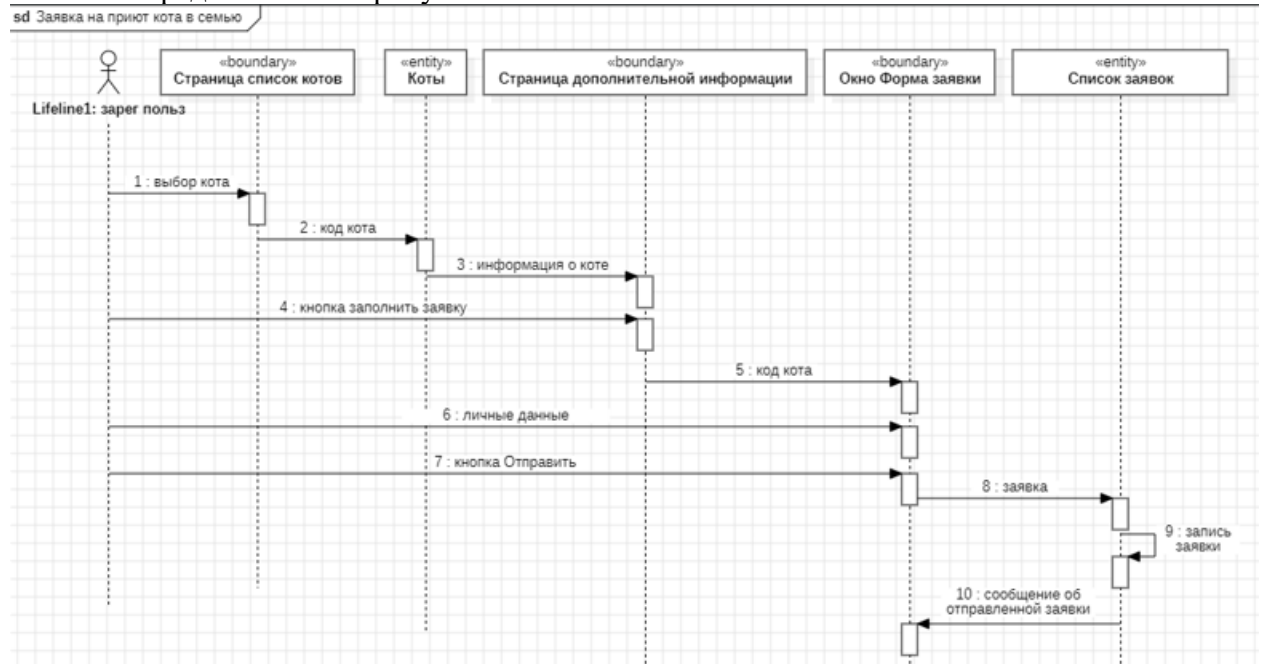


Рисунок 5 – Диаграмма последовательности

Зарегистрированный пользователь выбирает кота на страничке со списком котов, при переходе на страницу с дополнительной информацией, код кота поступает в таблицу скотами и из этой сущности уже передается информация на страницу с дополнительной информацией о коте. На этой странице пользователь может нажать кнопку “заполнить заявку”, тогда открывается окно “Форма заявки”, куда из странички с дополнительной информацией передается код выбранного питомца. Автоматические вводятся данные пользователя, и он нажимает кнопку отправить. Эта заявка передается в сущность Список заявок, там она записывается в общий список и пользователю выводится сообщение о том, что его заявка была отправлена.

### Динамические аспекты системы

На рисунке 6 диаграмма состояния объекта Заявка на приют кота в семью.



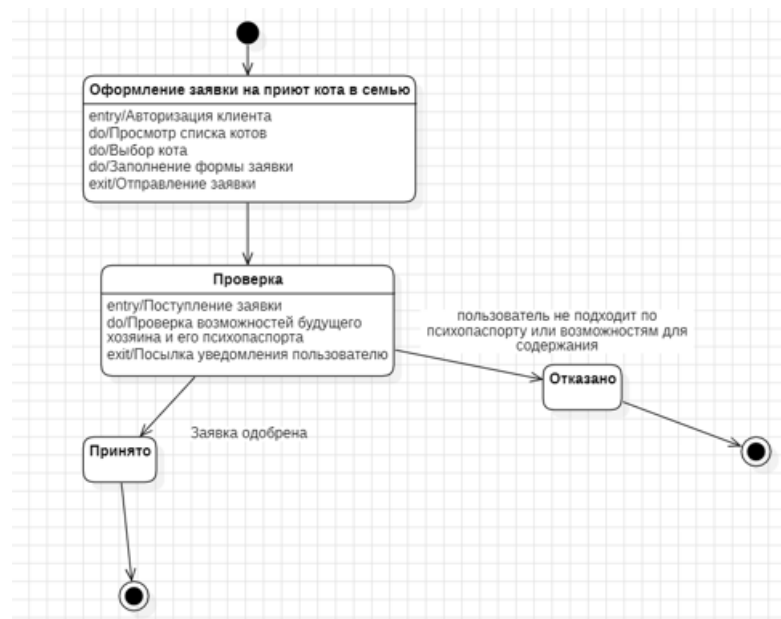


Рисунок 6 – Диаграмма состояния объекта Заявка на приют кота в семью

На рисунках 7-8 диаграмма активности для заявки на приют кота в семью.

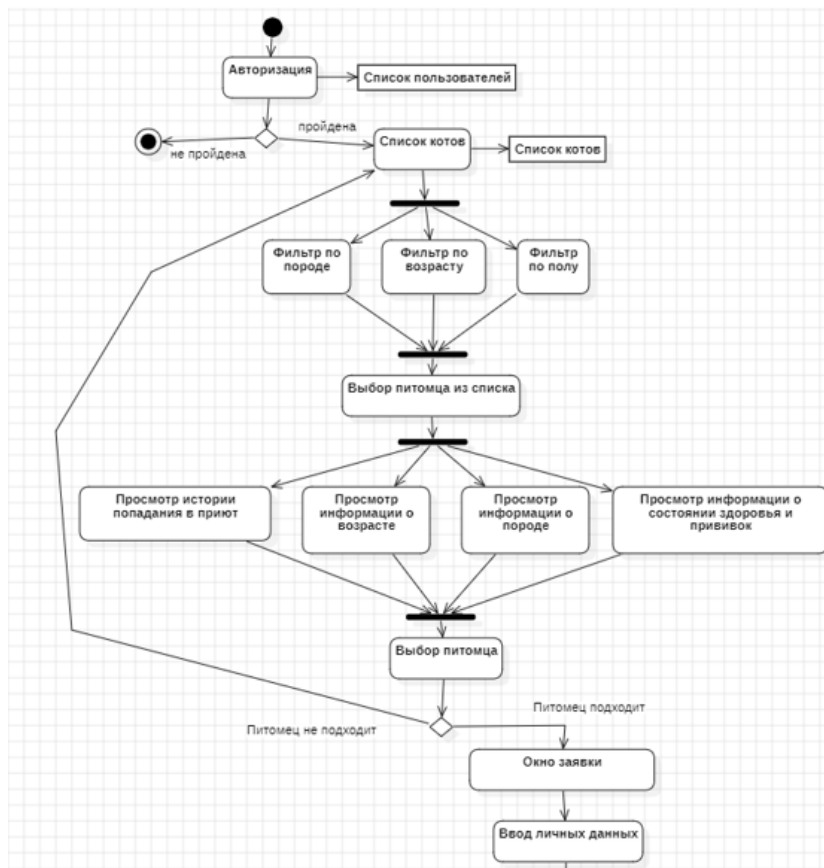


Рисунок 7 - Диаграмма активности заявки на приют кота в семью часть 1

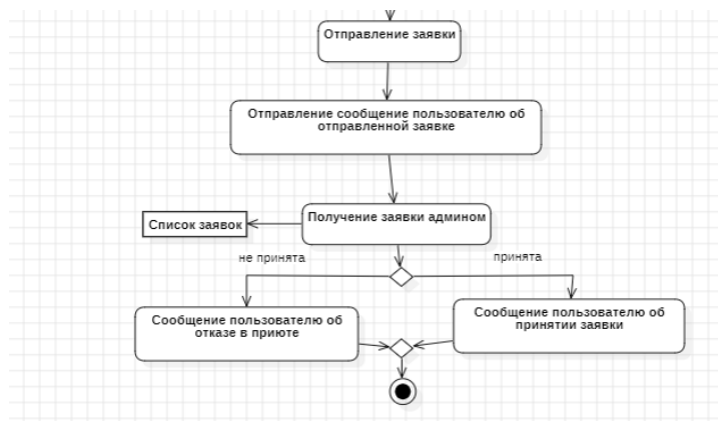


Рисунок 8 – Диаграмма активности заявки на приют кота в семью часть 2

На рисунке 9 диаграмма активности для заявки на пристройку кота в приют применением нескольких категорий пользователей.

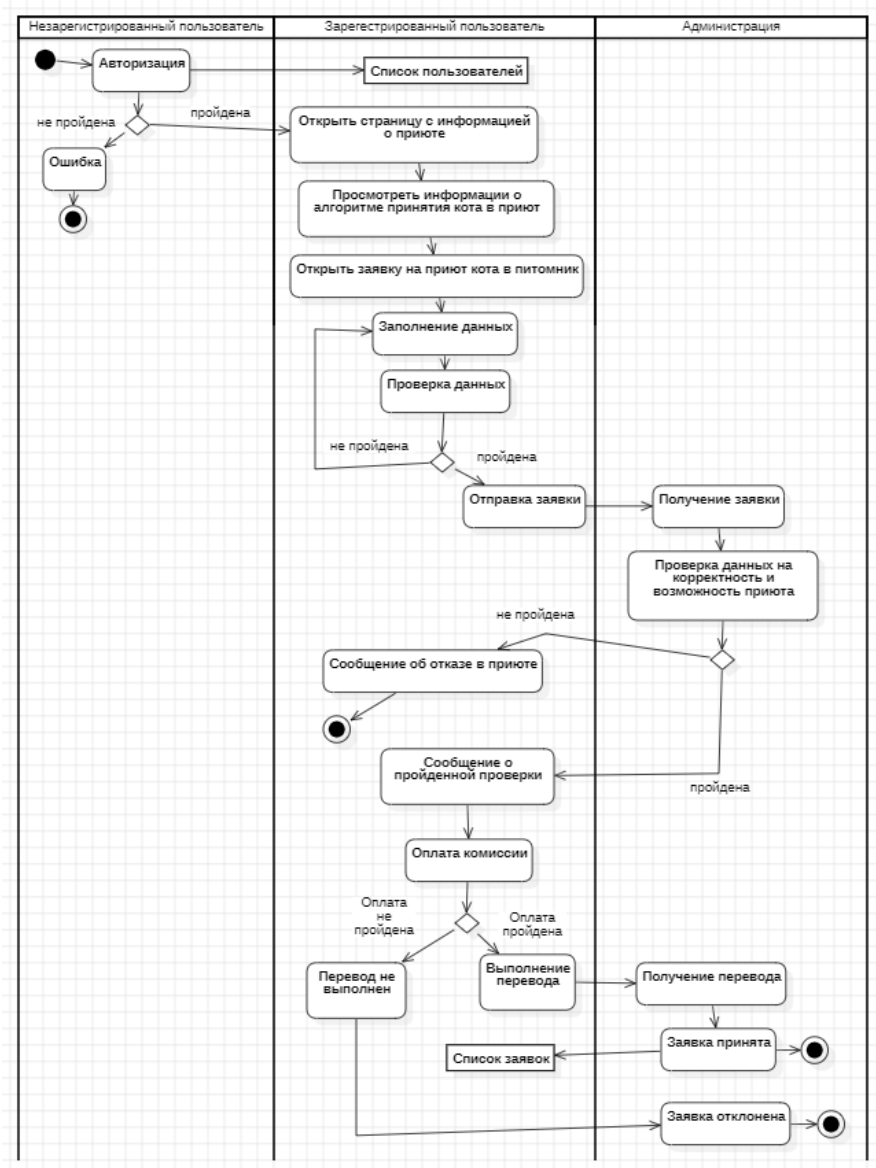


Рисунок 9 – Диаграмма активности заявки на пристройку кота в приют

# ПРОЕКТИРОВАНИЕ БД

## Сущности

Выделенные сущности:

- Кот
- Статус кота
- Фото
- Карточка болезней
- Болезнь
- Состояние кота
- Карточка прививок
- Прививка
- Порода
- Пользователь
- Волонтер
- Мероприятие
- Заявка на кота
- Тип заявки
- Статус заявки
- Заявка на мероприятие
- Оплата
- Тип оплаты
- Черный список пользователей

Описание сущностей:

1. Сущность “Кот” содержит в себе информацию о питомцах. Атрибуты: id, кличка, id породы, пол, дата рождения, причина попадания в приют, статус

кота относительно приюта.

2. Сущность “Статус кота” содержит в себе информацию о том, где в данный момент находится кот: в приюте, в семье, на передержке. Атрибуты: id, статус.

3. Сущность “Фото” содержит в себе информацию о фотографии кота. Атрибуты: id, фото (ссылка на фото), id кота и дата.

4. Сущность “Карточка болезней” содержит в себе информацию о здоровье питомца. Атрибуты: id, id кота, id болезни, дата заболевания, назначение ветеринара, дата выздоровления, id состояния кота.

5. Сущность “Болезнь” содержит в себе информацию о возможных болезнях. Атрибуты: id, название болезни.

6. Сущность “Состояние кота” содержит в себе информацию о том, в каком состоянии кот находится на данный момент относительно этой болезни. Атрибуты: id, состояние.

7. Сущность “Карточка прививок” содержит в себе информацию о прививках питомца. Атрибуты: id, id кота, id прививки, дата.

8. Сущность “Прививка” содержит в себе информацию о возможных прививках. Атрибуты: id, название.

9. Сущность “Порода” содержит в себе информацию о породах котов. Атрибуты: id, название.

10. Сущность “Пользователь” содержит в себе информацию о пользователе. Атрибуты: id, фамилия, имя, отчество, пол, дата рождения, почта, телефон, количество людей в семье, место работы, средний заработок, количество детей в семье, опыт с животными, id улицы, дом и квартира, площадь дома, логин, пароль.

12. Сущность “Волонтер” содержит в себе информацию о волонтерах приюта. Атрибут: id, фамилия, имя, отчество, пол, дата рождения, фото.

13. Сущность “Мероприятие” содержит в себе информацию о волонтерских мероприятиях. Атрибуты: id, название, дата, время, место, деятельность, id волонтера.

14. Сущность “Заявка на кота” содержит в себе информацию о заявке оставленной пользователем на кота, чтобы взять или отдать кота в приют. Атрибуты: id, id пользователя, id кота, id статус заявки, id тип заявки, примечание, дата.

15. Сущность “Тип заявки” содержит в себе информацию о типе заявки на кота: отдать в приют или взять. Атрибуты: id, тип.

16. Сущность “Статус заявки” содержит в себе информацию о статусе заявки от пользователя. Атрибуты: id, статус.

17. Сущность “Заявка на мероприятие” содержит в себе информацию о заявке, оставленной пользователем, на мероприятие. Атрибуты: id, id пользователя, id мероприятия, id статус заявки, дата.

18. Сущность “Оплата” содержит в себе информацию об оплате от пользователя. Атрибуты: id, id пользователя, сумма, дата и время, id тип оплаты.

19. Сущность “Тип оплаты” содержит в себе информацию о типе оплаты от пользователя. Оплата в качестве пожертвования или комиссии. Атрибуты: id, тип.

20. Сущность “ Черный список” содержит в себе информацию о пользователях, заявки от которых не будут обрабатываться. Атрибуты: id, id пользователя, причина, дата.

Связь 1 ко многим, not Nulls, identifying между сущностями:

- Болезнь – карточка болезней
- Состояние кота – карточка болезней
- Кот – карточка болезней
- Статус кота – кот
- Кот – карточка прививок
- Прививка – карточка прививок
- Порода – кот (Nulls Allowed)
- Кот – фото
- Кот – заявка на кота

- Статус заявки – заявка на кота
- Тип заявки – заявка на кота
- Статус заявки – заявка на мероприятие
- Пользователь – заявка на кота
- Пользователь – заявка на мероприятие
- Мероприятие – заявка на мероприятие
- Волонтер – мероприятие
- Пользователь – черный список
- Пользователь – оплата
- Тип оплаты – оплата

## Модели

Логическая модель на рисунке 10.

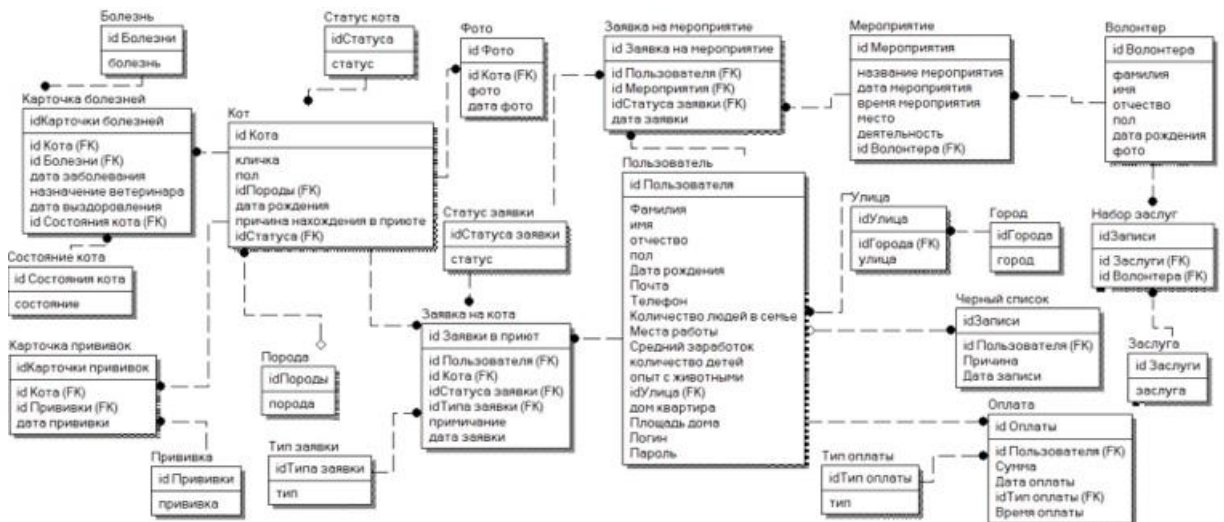


Рисунок 10- Логическая модель

Физическая модель на рисунке 11.

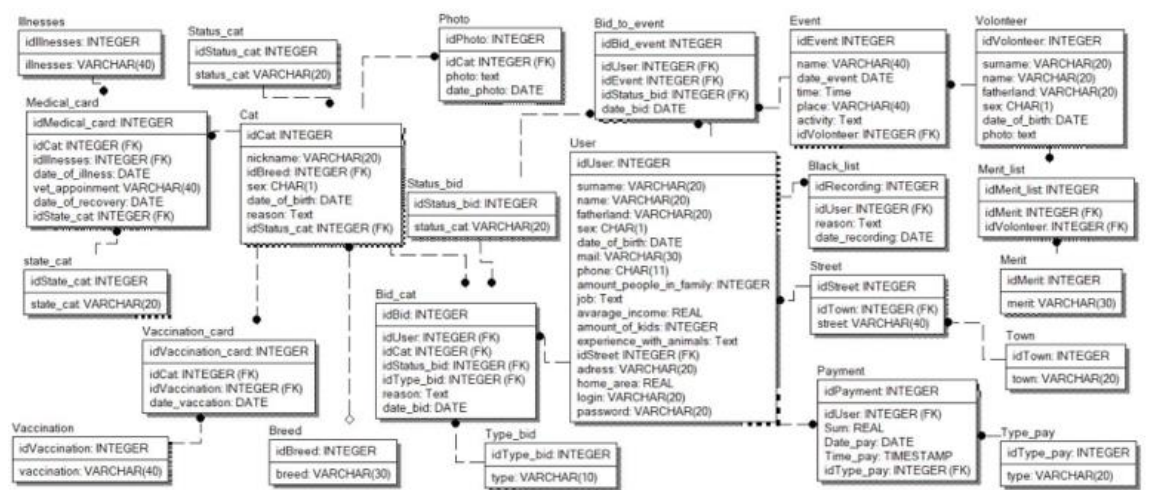


Рисунок 11 – Физическая модель  
Схема данных в MS Access на рисунке 12.

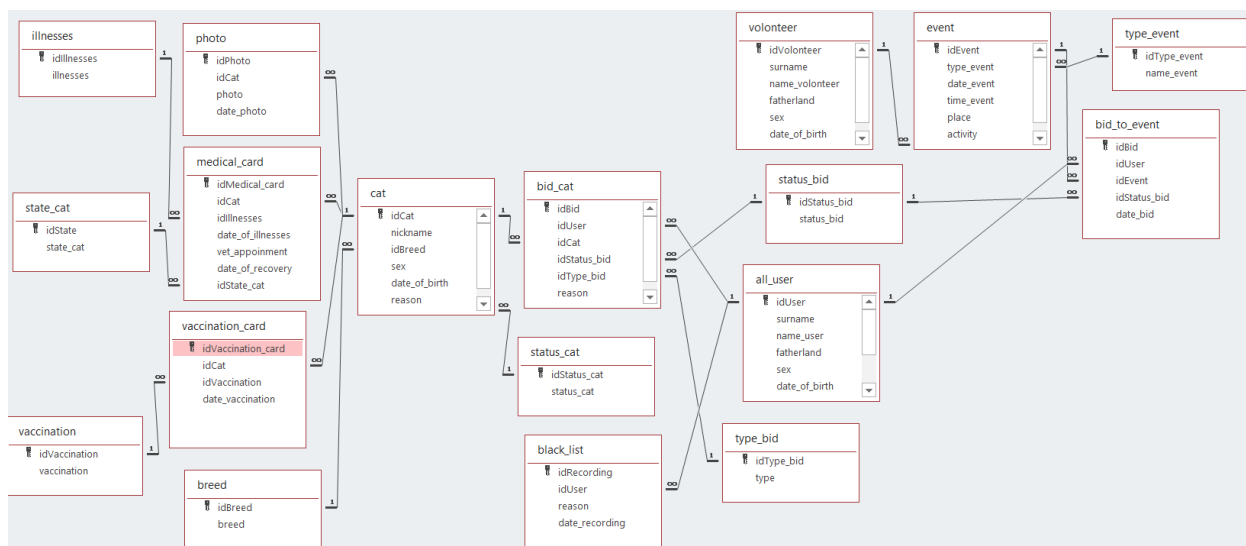


Рисунок 12 – Схема БД

# Создание сайта с использованием технологии ASP.NET

## Разметка сайта. Фильтрация данных

Страницы сайта на рисунках 1-16.



Рисунок 13 – Главная страница

Порода		Возраст		Пол		Котик	
Не выбрано ▾		Не выбрано ▾		Не выбрано ▾		Показать	<input type="text"/>
						Показать	Отменить
	ID	Кличка	Порода	Пол	Дата рождения	Причина нахождения в приюте	Фото
<a href="#">Выбор</a>	1	Пушок	Без породы	М	12.04.2022	нашли на улице волонтеры	
<a href="#">Выбор</a>	3	Арес	Рэгдолл	М	16.12.2021	аллергия	
<a href="#">Выбор</a>	4	Ника	Спамский	Ж	01.01.2004	родилась в приюте	
<a href="#">Выбор</a>	6	Соня	Као-Мани	Ж	20.05.2019	принесли	
<a href="#">Выбор</a>	7	Дэвнис	Рэгдолл	М	18.12.2020	родился в приюте	

Рисунок 14 – Страница с каталогом



Порода

Возраст

Пол

Котик

Не выбрано

От 1 до 3 лет

M

Показать

Показать

Отменить

	ID	Кличка	Порода	Пол	Дата рождения	Причина нахождения в приюте	Фото
<a href="#">Выбор</a>	1	Пушок	Без породы	M	12.04.2022	нашли на улице волонтеры	
<a href="#">Выбор</a>	3	Арес	Рэгдолл	M	16.12.2021	аллергия	
<a href="#">Выбор</a>	7	Дэвнис	Рэгдолл	M	18.12.2020	родился в приюте	

Рисунок 15 – Фильтры

Котик Арес

Порода

Пол

Дата рождения

Причина попадания в приют

Прививки

Болезни

Рэгдолл

M

16.12.2021

аллергия

Прививка	Дата вакцинации
панлейкопения	15.01.2022
кальцивироз	20.01.2022
хламидиоз	05.02.2022
бешенство	15.02.2022

Болезнь	Дата заболевания	Назначение врача	Дата выздоровления	Состояние
хламидиоз	15.06.2022	Диета	25.12.2022	здоров
ожирение	13.12.2023	таблетки		болен

Фото

Подать заявку

Рисунок 16 – Страница котика

Мероприятия

	ID	Название	Дата	Время	Место	Деятельность	Волонтер
<a href="#">Выбор</a>	1	Уход за животными	10.05.2022	15:00	Приют	Вычесать, помыть и поиграть с животными	Окунина
<a href="#">Выбор</a>	2	Поездка в ветклинику	05.07.2021	12:00	Ветклиника "Дружок"	Помочь побыть с животными	Васильева
<a href="#">Выбор</a>	3	Уход за животными	20.01.2024	17:00	Приют	Помощь по уходу	Окунина
<a href="#">Выбор</a>	5	Уборка в приюте	14.12.2023	14:00	Квартира с передержкой	Уборка в клетках	Доринина

Запись на мероприятие

Рисунок 17 – Страница с мероприятиями



Твой котик



Логин  
Пароль

[Главная](#)  
[Каталог питомцев](#)  
[Мероприятия](#)  
[Пожертвование](#)  
[Правила сдачи котика в приют](#)  
[Волонтеры](#)  
[Информация о приюте](#)

Фамилия  
Имя  
Отчество  
Мероприятие  
Дата

Label  
Label  
Label

Заявка на мероприятие

Рисунок 18 – Запись на мероприятие

Сделать пожертвование

Сумма

Рисунок 19 – Страница с пожертвованием

Правила сдачи кота в приют

1. Заполните заявку
  2. После принятия заявки оплатите пошлину и отправьте чек на почту приюте
  3. Принесите котика
- После оплаты пошлины мы с вами свяжемся

Рисунок 20 – Страница с правилами



Твой котик



Логин  
Пароль

[Главная](#)  
[Каталог питомцев](#)  
[Мероприятия](#)  
[Пожертвование](#)  
[Правила сдачи котика в приют](#)  
[Волонтеры](#)  
[Информация о приюте](#)

Фамилия	Имя	Отчество	Пол	Дата рождения	Фото
Окунина	Ирина	Ивановна	Ж	15.04.2005	
Доринина	Анна	Евгеньевна	Ж	27.03.2002	
Васильева	Полина	Константиновна	Ж	09.07.2004	

Наши волонтеры

Рисунок 21 – Страница с волонтерами



Твой котик



Логин:   
Пароль:

[Главная](#)  
[Каталог питомцев](#)  
[Мероприятия](#)  
[Пожертвования](#)  
[Правила сдачи котика в приют](#)  
[Волонтеры](#)  
[Информация о приюте](#)

Наш приют

Информация о приюте:

Контакты:  
Почта: Label  
Телефон: Label

Наш адрес: Label

## Рисунок 22 – Информация о приюте

### Данные

Фамилия: Ковалев  
Имя: Иван  
Отчество: Денисович  
Пол: М  
Дата рождения: 12.06.1984  
Почта: jghg@mail.ru  
Телефон: 85462156325  
Логин: kovalevan  
Пароль: 123123

### История пожертвований

Сумма	Дата	Время	Тип
5000	15.10.2022	13:12	комиссия
15000	22.11.2023	10:59	пожертвование

### История заявок

#### Заявки на кота

Имя котика	Статус заявки	Тип заявки	Причина отказа	Дата
Арес	принято	в семью		15.10.2022
Ника	принято	в семью		23.11.2023
Мушка	принято	в приют		28.11.2023

## Рисунок 23 – Профиль пользователя

Введите данные которых хотите изменить

Фамилия:   
Имя:   
Отчество:   
Пол:    
Дата рождения:    
Почта:   
Телефон:   
Логин:   
Пароль:

## Рисунок 24 – Изменение данных пользователя

Фамилия	<input type="text"/>
Имя	<input type="text"/>
Отчество	<input type="text"/>
Пол	<input type="text" value="Ж"/> ▼
Дата рождения	<input type="text" value="дд.мм.гггг"/> <input type="button" value="📅"/>
Почта	<input type="text"/>
Телефон	<input type="text"/>
Логин	<input type="text"/>
Пароль	<input type="password"/>
Повторите пароль	<input type="password"/>

Рисунок 25 – Страница с регистрацией

## Реализация личного кабинета пользователя

При регистрации, пользователь должен заполнить все поля. Рисунок 26.

Фамилия	<input type="text"/>	Заполните поле
Имя	<input type="text"/>	Заполните поле
Отчество	<input type="text"/>	
Пол	<input type="text" value="Ж"/> ▼	
Дата рождения	<input type="text" value="дд.мм.гггг"/> <input type="button" value="📅"/>	Заполните поле
Почта	<input type="text"/>	Заполните поле
Телефон	<input type="text"/>	Заполните поле
Логин	<input type="text"/>	Заполните поле
Пароль	<input type="password"/>	Заполните поле
Повторите пароль	<input type="password"/>	Заполните поле

Рисунок 26 – Ошибка заполнения

Код регистрации в приложении А

После этого у пользователя открывается доступ в личный кабинет (рисунок 27).



Рисунок 27 – Доступ в личный кабинет

Код входа в приложении Б.

Также есть ошибка при неправильном вводе логина или пароля (рисунок 28).

**Ошибка**

Логин

Пароль

Рисунок 28 – Ошибка входа

При подаче заявки пользователь должен находиться в системе (рисунок 29)

← → ↻ localhost:52113/pages/cat\_page.aspx

[Правила сдачи котика в приют](#)
[Волонтеры](#)
[Информация о приюте](#)


Дата рождения 12.04.2022  
 Причина попадания в приют нашли на улице волонтеры

Прививки

Прививка	Дата вакцинации
пандейкопения	12.07.2022
хламидиоз	20.07.2022

Болезни Нет болезней

Фото




Необходимо войти в учетную запись

Рисунок 29 – Ошибка

Теперь пользователь можно оставлять заявку на кота (рисунок 30).

← → ↻ localhost:52113/pages/order\_cat\_page.aspx



[Главная](#)  
[Каталог питомцев](#)  
[Мероприятия](#)  
[Пожертвование](#)  
[Правила сдачи котика в приют](#)  
[Волонтеры](#)  
[Информация о приюте](#)

**Твой котик**

Заявка на котика

Фамилия Ковалев

Имя Иван

Отчество Денисович

Котик Пушок

Рисунок 30 – Заявка на кота

Также есть проверка на то, что пользователь дважды отправил такую заявку. Код отправления заявки в приложении В.

Для того, чтобы сделать заявку на мероприятие, его необходимо выбрать (рисунок 31, 32).

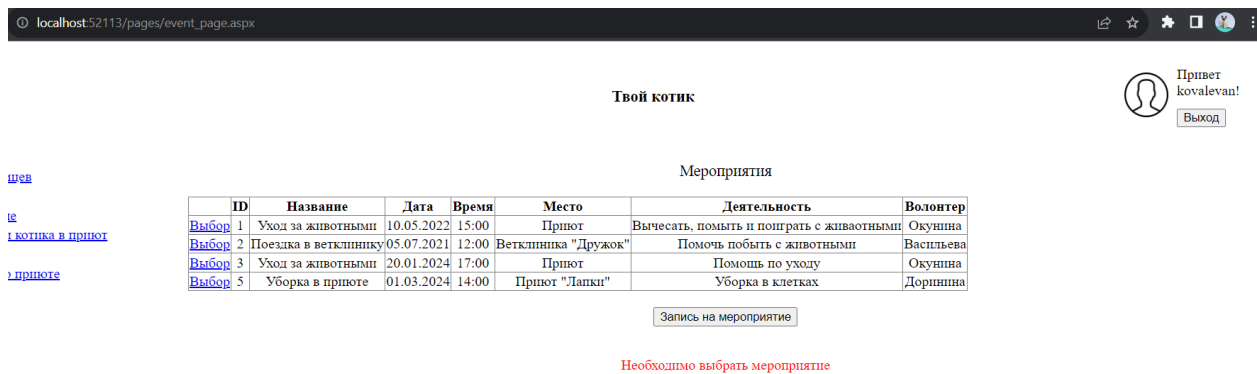


Рисунок 31 – Ошибка при заявке на мероприятие

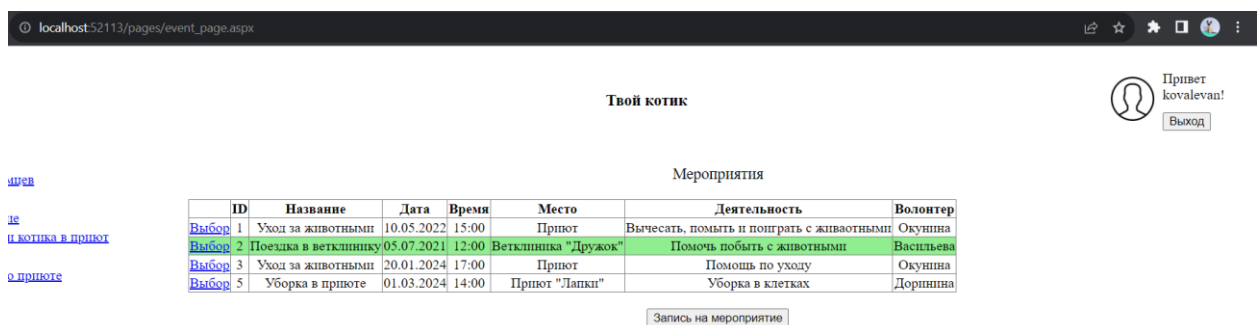


Рисунок 32 – Корректные действия для заявки на мероприятие

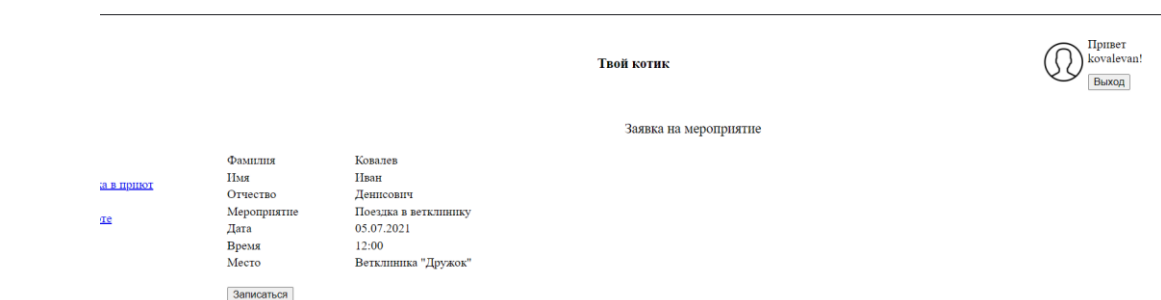


Рисунок 33 – Вид заявки на мероприятие

Код отправления заявки на мероприятие в листинге Г.

Пользователь может оставить заявку на то, чтобы сдать котика в приют. Для этого ему сначала надо зарегистрировать котика, его болезни и прививки (рисунки 34 - 36).

Твой котик

Заявка для сдачи котика в приют

Информация о котике

Кличка

Порода  если породы вашего котика нет в списке, то напишите ее

Пол

Дата рождения

Причина

Заболевания

Болезнь  или

Дата заболевания

Назначение врача

Дата выздоровления

Состояние

Рисунок 34 – Регистрация котика

Твой котик

Заявка для сдачи котика в приют

Информация о котике

Кличка

Порода  если породы вашего котика нет в списке, то напишите ее

Пол

Дата рождения

Причина  Заполните

Заболевания

Болезнь  или

Дата заболевания

Назначение врача

Дата выздоровления

Состояние

Сначала внесите информацию о котике

Рисунок 35 – Ошибка при регистрации котика

## Информация о котике

Кличка   
 Порода  если породы вашего котика нет в списке, то напишите ее   
 Пол   
 Дата рождения   
 Причина

## Заболевания

Болезнь  или   
 Дата заболевания   
 Назначение врача   
 Дата выздоровления   
 Состояние

Заболевание	Дата заболевания	Назначение врача	Дата выздоровления	Состояние
Блохи	29.12.2023	Таблетки		Болен

## Прививки

Прививка  или   
 Дата прививки

Рисунок 36 – Внесение информации о котике  
Код для сдачи котика в приют в приложении Д.

В личном кабинете пользователя есть информация о пользователе (рисунок 37)

## Твой котик

## Личный кабинет

## Данные

Фамилия	Ковалев
Имя	Иван
Отчество	Денисович
Пол	М
Дата рождения	12.06.1984
Почта	jjghg@mail.ru
Телефон	85462156325
Логин	kovalevan
Пароль	123123

## История пожертвований

Сумма	Дата	Время	Тип
5000	15.10.2022	13:12	комиссия
15000	22.11.2023	10:59	пожертвование

## История заявок

## Заявки на кота

Имя котика	Статус заявки	Тип заявки	Причина отказа	Дата
Арес	принято	в семью		15.10.2022

Рисунок 37 – Личный кабинет  
Пользователь может изменить свои личные данные (рисунок 38).



Изменить данные

Введите данные которых хотите изменить

Фамилия	<input type="text"/>
Имя	<input type="text"/>
Отчество	<input type="text"/>
Пол	<input type="text" value="М"/>
Дата рождения	<input type="text" value="дд.мм.гггг"/>
Почта	<input type="text"/>
Телефон	<input type="text"/>
Логин	<input type="text"/>
Пароль	<input type="text"/>

Изменить

## Рисунок 38 – Изменение личных данных

Код для отображения страницы пользователя и изменение данных в приложении Е.

## Администрирование

Страница администратора открывается через кнопку “Управление”. И открывается страница с разными разделами (рисунок 39).



[Главная](#)  
[Каталог питомцев](#)  
[Мероприятия](#)  
[Пожертвование](#)  
[Правила сдачи котика в приют](#)  
[Волонтеры](#)  
[Информация о приюте](#)

[Котики](#)  
[Мероприятия](#)  
[Заявки на котиков](#)

Твой котик



Привет admin!

Управление

Выход

## Рисунок 39 – Главная страница админа

Страница с котиками на рисунке 40. Есть различные фильтры.

Твой котик

Порода	Возраст	Пол	Статус	
<input type="text" value="Не выбрано"/>	<input type="text" value="Не выбрано"/>	<input type="text" value="Не выбрано"/>	<input type="text" value="Не выбрано"/>	<input type="button" value="Показать"/>
Котик	<input type="text"/>	<input type="button" value="Показать"/>	<input type="button" value="Отменить"/>	

Котик

id	Кличка	Порода	Пол	Дата рождения	Причина	Статус	Фото
<a href="#">Выбор</a> 1	Пушок	Без породы	М	12.04.2022	нашли на улице волонтеры	приют	
<a href="#">Выбор</a> 3	Арес	Рыболов	М	16.12.2021	аллергия	приют	
<a href="#">Выбор</a> 4	Ника	Синий	Ж	01.01.2004	родилась в приюте	приют	
<a href="#">Выбор</a> 5	Энне	Без породы	М	14.08.2018	попала в приют с улицы	перепелка	

## Рисунок 40 – Страница с котиками


Выбранному котика можно изменить статус нахождения (рисунок 41).

**Твой котик**

Порода  Возраст  Пол  Статус

Котик


Котик

id	Кличка	Порода	Пол	Дата рождения	Причина	Статус	Фото	
<a href="#">Выбор</a>	3	Арес	Разгол	М	16.12.2021	аллергия	привот	

Изменить статус кота

Новый статус

Фото

Кличка	Фото	Дата
Арес		14.05.2022

**Рисунок 41 – Выбор котика**  
 Также ему можно добавить болезнь, и в целом создать новую (рисунок 42).

**Болезни**

Состояние:

	id	Кличка	Болезнь	Дата заболевания	Назначение врача	Дата выздоровления	Состояние
<a href="#">Выбор</a>	1	Арес	хламидиоз	15.06.2022	Днепа	25.12.2022	здоров
<a href="#">Выбор</a>	29	Арес	ожирение	13.12.2023	таблетки		болен

Добавить болезнь котiku

Выберите котика в таблице сверху

Болезнь

Дата заболевания

Назначение врача

Название

Добавить болезнь

**Рисунок 42 – Взаимодействие с болезнями**  
 Можно изменить состояние болезни (рисунок 43).

Болезнь

id	Кличка	Болезнь	Дата заболевания	Назначение врача	Дата выздоровления	Состояние
29	Арес	ожирение	13.12.2023	таблетки		болен

Дата выздоровления

Состояние

**Рисунок 43 – Изменение состояние котика и дата выздоровления**  
 Также есть возможность добавить новую прививку котiku. И в целом создать новый вид прививки (рисунок 44).

Кличка	Прививка	Дата
Арес	панлейкопения	15.01.2022
Арес	кальцивироз	20.01.2022
Арес	хламидиоз	05.02.2022
Арес	бешенство	15.02.2022

Прививки

Добавить прививку коту

Выберите котика в таблице сверху

Прививка

Дата прививки

Добавить

Название

Добавить прививку

Добавить

## Рисунок 44 – Взаимодействие с прививками

На всех кнопках стоит проверка, чтобы котик был выбран (рисунок 45).

Выберите котика в таблице сверху

Болезнь

Дата заболевания

Назначение врача

Добавить

Сначала выберите котика

## Рисунок 45 – Проверка выбора котика

Код данной странички в приложении Ж.

На странице мероприятия можно редактировать выбранное или создать новое (рисунок 46).

Мероприятия

Статус:

	id	Тип	Дата	Время	Место	Деятельность	Волонтер
<a href="#">Выбор</a>	1	Уход за животными	10.05.2022	15:00	Приют	Вычесать, помыть и пограть с животными	Окунина
<a href="#">Выбор</a>	2	Поездка в ветклинику	05.07.2021	12:00	Ветклиника "Дружок"	Помочь побыть с животными	Васильева
<a href="#">Выбор</a>	3	Уход за животными	20.01.2024	17:00	Приют	Помощь по уходу	Окунина
<a href="#">Выбор</a>	5	Уборка в приюте	14.12.2023	14:00	Квартира с передержкой	Уборка в клетках	Доринина

Редактировать мероприятие

Дата

Время  В формате чч:мм

Место

Волонтер

Изменить

Создание нового мероприятия

Название

Дата

Время  В формате чч:мм

Место

Деятельность

Фамилия

Добавить

## Рисунок 46 – Работа с мероприятиями

Код данной странички в приложении З.

В заявках на котика можно отсортировать заявки, выбрать заявку и просмотреть информацию о человеке из заявки (рисунок 47).

Заявки на кота

Тип

Не выбрано

Статус

Не выбрано

Показать

Отмена

	id заявки	id пользователя	Кличка котика	Статус заявки	Тип заявки	Причина	Дата
<a href="#">Выбор</a>	1	1	Арес	принято	в семью		15.10.2022
<a href="#">Выбор</a>	2	2	Соня	ожидание	в семью		12.09.2023
<a href="#">Выбор</a>	3	5	Арес	отклонено	в семью	не прошел проверку врача	02.03.2022
<a href="#">Выбор</a>	4	5	Соня	принято	в приют	не нужна	01.07.2019
<a href="#">Выбор</a>	5	4	Эрнес	принято	на передержку		02.10.2023
<a href="#">Выбор</a>	6	1	Ника	принято	в семью		23.11.2023
<a href="#">Выбор</a>	7	1	Мушка	принято	в приют		28.11.2023

Информация о человеке

Показать

Скрыть

id	Фамилия	Имя	Отчество	Пол	Дата рождения	Почта	Телефон
4	Нижний	Валерий	Александрович	М	29.11.2000	dbbfsjb@mail.ru	84521567854

Рисунок 47 – Заявка на котика

Также можно изменить статус заявки и на мероприятие, и на котика (рисунок 48).

Изменить статус заявки

принято

Причина отказа:

Изменить

Заявки на мероприятия

Статус

принято

Показать

Отмена

	id заявки	id пользователя	Мероприятие	Статус	Дата
<a href="#">Выбор</a>	1	1	Поездка в ветклинику	принято	05.05.2022
<a href="#">Выбор</a>	2	2	Уход за животными	принято	04.02.2022
<a href="#">Выбор</a>	3	4	Уход за животными	отклонено	10.05.2022
<a href="#">Выбор</a>	4	5	Уборка в приюте	ожидание	12.12.2023
<a href="#">Выбор</a>	5	2	Уборка в приюте	ожидание	10.11.2023
<a href="#">Выбор</a>	12	1	Уход за животными	ожидание	22.11.2023

Изменить статус заявки

принято

Изменить

Рисунок 48 – Заявки на мероприятие

Код данной странички в приложении И.

## **Заключение**

По итогу курсовой работы была разработана модель сайта для питомника, в котором содержатся кошки. Благодаря данному сайту, люди могут подробно узнать о котах, которые в поисках своего дома, и в случае чего забрать одного или нескольких к себе. Также пользователь сможет пристроить кота в приют, узнать о мероприятиях, поддерживающих приют, и поучаствовать в них вместе с волонтерами, а также отправить пожертвования.

```
// регистрация пользователя
protected void regestration(object sender, EventArgs e)
{
    if (Page.IsValid)
    {
        try
        {
            string connectionString =
"Provider=Microsoft.ACE.OLEDB.12.0; Data Source=d:\\PIS\\DatabaseCat.accdb";
            var connect = new OleDbConnection(connectionString);
            connect.Open();
            string ssurname, sname, sfather, ssex, sbirth, smail, sphone,
slogin, spassword;
            ssurname = Convert.ToString(TextBox_surname.Text);
            sname = Convert.ToString(TextBox_name.Text);
            sfather = Convert.ToString(TextBox_father.Text);
            ssex = List_sex.SelectedValue;
            sbirth = Convert.ToString(TextBox_date_birth.Text);
            smail = Convert.ToString(TextBox_mail.Text);
            sphone = Convert.ToString(TextBox_phone.Text);
            slogin = Convert.ToString(TextBox_login.Text);
            spassword = Convert.ToString(TextBox_password.Text);
            var mycom = new OleDbCommand();
            mycom.Connection = connect;
            string sqlQ;
            sqlQ = "INSERT INTO all_user(surname, name_user, fatherland,
sex, date_of_birth, mail, phone, login, [password]) " +
"VALUES ('" + ssurname + "', '" + sname + "', '" + sfather
+ "', '" + ssex + "', '" + sbirth + "', '" +
smail + "', '" + sphone + "', '" + slogin + "', '" +
spassword + "');"
            mycom.CommandText = sqlQ;
            mycom.ExecuteNonQuery();
            connect.Close();
            Error_regestration.Text = "Регистрация прошла успешно";
        }
        catch
        {
            Error_regestration.Text = "Ошибка";
        }
    }
}
```

```
// вход в систему пользователем
protected void Entry(object sender, EventArgs e)
{
    string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
    var connect = new OleDbConnection(connectionString);
    connect.Open();
    string slogin, sparol;
    slogin = Convert.ToString(Enter_login.Text);
    sparol = Convert.ToString(Enter_password.Text);
    // поиск по базе
    string sqlQ = "SELECT * FROM all_user WHERE login='" + slogin + "' AND
password='" + sparol + "'";
    var mycom = new OleDbCommand();
    mycom.CommandText = sqlQ;
    mycom.Connection = connect;
    OleDbDataReader myread;
    myread = mycom.ExecuteReader();

    if (myread.Read() == true)
    {
        Session["IdU"] = slogin;
        Session["IDP"] = Convert.ToString(myread.GetValue(0));
        Enter_login.Visible = false;

        Error.Visible = false;
        Button_entry.Visible = false;
        Button_registration.Visible = false;
        Button_exit.Visible = true;
        Data.Visible = true;
        ImageButton_profile.Visible = true;
        Data.Text = "Привет " + slogin + "!";
        if (slogin == "admin" && sparol == "admin")
            Bt_Admin.Visible = true;
    }
    else
    {
        Error.Visible = true;
        Error.Text = "Ошибка";
    }
    connect.Close();
}

// выход из системы
protected void Exit(object sender, EventArgs e)
{
    Session.Clear();
    Enter_login.Visible = true;
    Enter_password.Visible = true;
    Error.Visible = false;
    Data.Visible = false;
    Text_login.Visible = true;
    Text_password.Visible = true;
    Button_entry.Visible = true;
    Button_registration.Visible = true;
    Button_exit.Visible = false;
}
```

```
// отправка заявки на кота в семью
protected void Order_cat(object sender, EventArgs e)
{
    long iduser = Convert.ToInt64(Session["IDP"]);
    int idcat = Convert.ToInt16(Session["IdC"]);
    string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
    var connect = new OleDbConnection(connectionString);
    connect.Open();
    var mycom = new OleDbCommand();
    mycom.Connection = connect;
    string sqlQ;
    // проверка на то, что такой заявки еще нет от пользователя
    sqlQ = "SELECT * FROM bid_cat where idCat=" + idcat + " and idUser=" +
iduser;
    DataOrderCat.SelectCommand = sqlQ;
    TableOrderCat.DataBind();
    if (TableOrderCat.Rows.Count == 0)
    {
        string curDate = DateTime.Now.ToString("yyyy-MM-dd");

        sqlQ = "INSERT INTO bid_cat(idUser, idCat, idStatus_bid, idType_bid,
date_bid) " +
        "VALUES (" + iduser + ", " + idcat + ", 3, 2, '" + curDate +
        "')";
        mycom.CommandText = sqlQ;
        mycom.ExecuteNonQuery();
        connect.Close();

        Label_button_order_cat.Text = "Заявка подана";
        Label_button_order_cat.Visible = true;
    }
    else
    {
        Label_button_order_cat.Text = "Такая заявка уже есть";
        Label_button_order_cat.Visible = true;
    }
}
}
```



```

        // проверки перед переходом на другую страницу
protected void Order_event(object s, EventArgs e)
{
    long iduser = Convert.ToInt64(Session["IDP"]);
    int idevent = Convert.ToInt16(Session["IdE"]);
    if (iduser == 0)
    {
        Label_error_order_event.Text = "Необходимо войти в учетную запись";
        Label_error_order_event.Visible = true;
    }
    else
    {
        if(idevent == 0)
        {
            Label_error_order_event.Text = "Необходимо выбрать мероприятие";
            Label_error_order_event.Visible = true;
        }
        else
        {
            Response.Redirect("order_event.aspx");
        }
    }
}

// выбор мероприятия
protected void TableEvent_SelectedIndexChanged(object sender, EventArgs e)
{
    for (int i=0; i<TableEvent.Rows.Count; i++)
        TableEvent.Rows[i].BackColor = System.Drawing.Color.White;
    int idrow = Convert.ToInt16(TableEvent.SelectedIndex);
    TableEvent.Rows[idrow].BackColor = System.Drawing.Color.LightGreen;
    int idevent = Convert.ToInt16(TableEvent.Rows[idrow].Cells[1].Text.ToString());
    Session["IdE"] = idevent;
}

// отправка заявки на мероприятие
protected void Order_event(object sender, EventArgs e)
{
    try
    {
        long iduser = Convert.ToInt64(Session["IDP"]);
        int idevent = Convert.ToInt16(Session["IdE"]);
        string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;
Data Source=d:\\PIS\\DatabaseCat.accdb";
        var connect = new OleDbConnection(connectionString);
        connect.Open();
        var mycom = new OleDbCommand();
        mycom.Connection = connect;
        string sqlQ;
        // проверка на повторения заявки
        sqlQ = "SELECT * FROM bid_to_event where idEvent=" + idevent + "
and idUser=" + iduser;
        bid_event.SelectCommand = sqlQ;
        TableBid_event.DataBind();
    }
}

```

```

        if (TableBid_event.Rows.Count == 0)
        {
            string curDate = DateTime.Now.ToString("yyyy-MM-dd");

            sqlQ = "INSERT INTO bid_to_event(idUser, idEvent, idStatus_bid,
date_bid) " +
                "VALUES (" + iduser + ", " + idevent + ", 3, '" + curDate +
                "')";

            mycom.CommandText = sqlQ;
            mycom.ExecuteNonQuery();
            connect.Close();

            Label_result_order_event.Text = "Заявка подана";
            Label_result_order_event.Visible = true;
        }
        else
        {
            Label_result_order_event.Text = "Такая заявка уже есть";
            Label_result_order_event.Visible = true;
        }
    }
    catch
    {
        Label_result_order_event.Text = "Ошибка";
        Label_result_order_event.Visible = true;
    }
}

```

```

        // создание записи о котике в бд
protected void Create_cat(object sender, EventArgs e)
{
    string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
    var connect = new OleDbConnection(connectionString);
    connect.Open();
    string name_cat, breed, sex, birth, reason;
    int idbreed;
    name_cat = Convert.ToString(TextBox_cat_name.Text);
    var mycom = new OleDbCommand();
    mycom.Connection = connect;

    if (ListBreed.SelectedValue == "0")
    {
        // выбор породы из поля
        breed = TextBox_breed.Text;
        string newbreed = "Insert into breed(breed) values('" + breed +
        "')";

        mycom.CommandText = newbreed;
        mycom.ExecuteNonQuery();
        connect.Close();
        // обновление таблицы
        TableBreed.DataSourceID = "all_breed";
        TableBreed.DataBind();
        connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
        connect = new OleDbConnection(connectionString);
        connect.Open();
        mycom = new OleDbCommand();
        mycom.Connection = connect;
    }
    else
    {
        // выбор породы из списка
        breed = ListBreed.SelectedValue;
    }
    string sql_breed = "SELECT * from breed where breed='" + breed + "'";
    all_breed.SelectCommand = sql_breed;
    TableBreed.DataBind();
    idbreed =
    Convert.ToInt16(TableBreed.Rows[0].Cells[0].Text.ToString());
    sex = ListSex.SelectedValue;
    birth = TextBox_birth.Text;
    reason = Convert.ToString(TextBox_reason.Text);

    // добавление котика в БД
    string sqlQ;
    sqlQ = "INSERT INTO cat(nickname, idBreed, sex, date_of_birth,
reason, idStatus_cat) " +
        "VALUES ('" + name_cat + "', " + idbreed + ", '" + sex + "', '"
+ birth + "', '" + reason + "', 5)";
    mycom.CommandText = sqlQ;
    mycom.ExecuteNonQuery();
    connect.Close();
    Label_craete_cat.Text = "Котик зарегистрирован";
    TableCat.DataSourceID = "DataCat";
    TableCat.DataBind();

    Label_craete_cat.Visible = true;
    Session["IdC"] = Convert.ToInt16(TableCat.Rows[TableCat.Rows.Count-
1].Cells[0].Text.ToString());

```

```

// добавление в бд болезней котика
protected void Add_illnesses(object sender, EventArgs e)
{
    int idcat = Convert.ToInt16(Session["IdC"]);
    if (idcat == 0)
    {
        Label_error_ill.Visible = true;
    }
    else
    {
        // подключение
        string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
        var connect = new OleDbConnection(connectionString);
        connect.Open();
        int idill, idstate_cat;
        string ill, date_ill, vet_app, date_rec, state_cat;
        var mycom = new OleDbCommand();
        mycom.Connection = connect;
        // осоздание новой болезни
        if (ListIllnesses.SelectedValue == "0")
        {
            // добавление болезни в бд
            ill = TextBox_illnesses.Text;
            string newill = "INSERT INTO illnesses(illnesses) VALUES (' +
ill + "')";
            mycom.CommandText = newill;
            mycom.ExecuteNonQuery();
            connect.Close();

            illnesses.DataSourceID = "illness";
            illnesses.DataBind();

            connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
            connect = new OleDbConnection(connectionString);
            connect.Open();
            mycom = new OleDbCommand();
            mycom.Connection = connect;

        }
        else
        {
            // выбор болезни из списка
            ill= ListIllnesses.SelectedValue;

            // поиск id этой болезни
            string sql_ill = "SELECT * from illnesses where illnesses='\" +
ill + \"'";
            illness.SelectCommand = sql_ill;
            illnesses.DataBind();
            idill = illnesses.Rows[0].Cells[0].Text.ToString();
            Convert.ToInt16(idill);
            // получение всех полей
            date_ill = TextBox_date_ill.Text;
            vet_app = TextBox_vet_app.Text;
            date_rec = TextBox_date_rec.Text;
            state_cat = ListStateCat.SelectedValue;

```

```

        string sql_state = "SELECT * from state_cat where state_cat='" +
state_cat + "'";
        Sqlstatecat.SelectCommand = sql_state;
        TableSateCat.DataBind();
        idstate_cat =
Convert.ToInt16(TableSateCat.Rows[0].Cells[0].Text.ToString());
        // добавление записи о болезни котика в БД
        string sqlQ;
        if (date_rec != "")
        {
            sqlQ = "INSERT INTO medical_card(idCat, idIllnesses,
date_of_illnesses, vet_appointment, date_of_recovery, idState_cat) " +
"VALUES (" + idcat + ", " + idill + ", '" + date_ill + "', '" +
vet_app + "', '" + date_rec + "', " + idstate_cat + ")";
        }
        else
        {
            sqlQ = "INSERT INTO medical_card(idCat, idIllnesses,
date_of_illnesses, vet_appointment, idState_cat) " +
"VALUES (" + idcat + ", " + idill + ", '" + date_ill + "', '" +
vet_app + "', " + idstate_cat + ")";
        }

        mycom.CommandText = sqlQ;
        mycom.ExecuteNonQuery();
        connect.Close();

        // отображение добавленных записей
        string sql_illnesses_new_cat = "SELECT * from medical_card where idCat="
+ idcat;
        DataMedicalCard.SelectCommand = sql_illnesses_new_cat;
        Table_illnesses.DataBind();
        Table_illnesses.Visible = true;

        // обнуление полей
        TextBox_illnesses.Text = "";
        TextBox_date_ill.Text = "";
        TextBox_vet_app.Text = "";
        TextBox_date_rec.Text = "";
        ListIllnesses.SelectedIndex = 0;
        ListStateCat.SelectedIndex = 0;
    }
    // Добавление прививок котiku
    protected void Add_vaccination(object sender, EventArgs e)
    {
        int idcat = Convert.ToInt16(Session["IdC"]);
        if (idcat == 0)
        {
            Label_error_vaccin.Visible = true;
        }
        else
        {
            // подключение
            string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
            var connect = new OleDbConnection(connectionString);
            connect.Open();
            int idvac;
            string vac, date_vac;
            var mycom = new OleDbCommand();
            mycom.Connection = connect;

```

```

        // осоздание новой прививки
if (List_vaccination.SelectedValue == "0")
{
    // добавление прививки в бд
    vac = Convert.ToString(TextBox_input_vaccin.Text);
    string new_vac = "INSERT INTO vaccination(vaccination) VALUES
('" + vac + "')";
    mycom.CommandText = new_vac;
    mycom.ExecuteNonQuery();
    connect.Close();

    TableVaccination.DataSourceID = "DataVaccination";
    TableVaccination.DataBind();
    connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
    connect = new OleDbConnection(connectionString);
    connect.Open();
    mycom = new OleDbCommand();
    mycom.Connection = connect;
}
else
{
    // выбор вакцины из списка
    vac = List_vaccination.SelectedValue;
}
// поиск id этой прививки
string sql_vac = "SELECT * from vaccination where vaccination='"
+ vac + "'";
DataVaccination.SelectCommand = sql_vac;
TableVaccination.DataBind();
idvac =
Convert.ToInt16(TableVaccination.Rows[0].Cells[0].Text.ToString());

    date_vac = TextBox_date_vaccin.Text;

    string sqlQ;
    sqlQ = "INSERT INTO vaccination_card(idCat, idVaccination,
date_vaccination) " +
        "VALUES (" + idcat + ", " + idvac + ", '" + date_vac + "')";

    mycom.CommandText = sqlQ;
    mycom.ExecuteNonQuery();
    connect.Close();

    // отображение добавленных записей
    string sql_vaccin_new_cat = "SELECT * from vaccination_card where
idCat=" + idcat;
    DataVaccinationCard.SelectCommand = sql_vaccin_new_cat;
    Table_vaccination_card.DataBind();
    Table_vaccination_card.Visible = true;

    // обнуление полей
    TextBox_date_vaccin.Text = "";
    TextBox_input_vaccin.Text = "";
    List_vaccination.SelectedIndex = 0;
}

```

```

// отправление заявки
protected void Send_order(object sender, EventArgs e)
{
    int idcat = Convert.ToInt16(Session["IdC"]);
    if (idcat == 0)
    {
        Label_message_order_in_shelter.Text = "Сначала внесите информацию о котике";
        Label_message_order_in_shelter.Visible = true;
    }
    else
    {
        long iduser = Convert.ToInt64(Session["IDP"]);

        // подключение
        string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data Source=d:\\PIS\\DatabaseCat.accdb";
        var connect = new OleDbConnection(connectionString);
        connect.Open();
        var mycom = new OleDbCommand();
        mycom.Connection = connect;
        string curDate = DateTime.Now.ToString("yyyy-MM-dd");
        string sqlQ;
        sqlQ = "INSERT INTO bid_cat(idUser, idCat, idStatus_bid, idType_bid, date_bid) " +
            "VALUES (" + iduser + ", " + idcat + ", 3, 1, '" + curDate +
            "')";
        mycom.CommandText = sqlQ;
        mycom.ExecuteNonQuery();
        connect.Close();
        Label_message_ok_cat.Visible = true;
    }
}

```

```
//Изменение данных пользователем
protected void Change_data(object sender, EventArgs e)
{
    long iduser = Convert.ToInt64(Session["IDP"]);

    string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
    var connect = new OleDbConnection(connectionString);
    connect.Open();
    var mycom = new OleDbCommand();
    mycom.Connection = connect;
    string sqlQ;
    if (TextBox_surname.Text != "")
    {
        sqlQ = "UPDATE all_user SET surname='" + TextBox_surname.Text + "'
where idUser="+iduser;
        mycom.CommandText = sqlQ;
        mycom.ExecuteNonQuery();
    }
    if (TextBox_name.Text != "")
    {
        sqlQ = "UPDATE all_user SET name_user='" + TextBox_name.Text + "'
where idUser=" + iduser;
        mycom.CommandText = sqlQ;
        mycom.ExecuteNonQuery();
    }
    if (TextBox_fatherland.Text != "")
    {
        sqlQ = "UPDATE all_user SET fatherland='" + TextBox_fatherland.Text
+ "' where idUser=" + iduser;
        mycom.CommandText = sqlQ;
        mycom.ExecuteNonQuery();
    }

    //// пропущен аналогичный код
    connect.Close();
    Response.Redirect("user_page.aspx");

    // отображение данных о пользователе
    protected void Page_Load(object sender, EventArgs e)
    {
        Session["IdC"] = 0;
        Session["IdE"] = 0;
        long iduser = Convert.ToInt64(Session["IDP"]);
        string str_user = "SELECT * FROM all_user where idUser=" + iduser;
        user.SelectCommand = str_user;
        TableUser.DataBind();

        Label_surname.Text = TableUser.Rows[0].Cells[1].Text.ToString();
        Label_name.Text = TableUser.Rows[0].Cells[2].Text.ToString();
        Label_father.Text = TableUser.Rows[0].Cells[3].Text.ToString();
        Label_sex.Text = TableUser.Rows[0].Cells[4].Text.ToString();
        Label_date_birth.Text = TableUser.Rows[0].Cells[5].Text.ToString();
        Label_mail.Text = TableUser.Rows[0].Cells[6].Text.ToString();
        Label_phone.Text = TableUser.Rows[0].Cells[7].Text.ToString();
        Label_login.Text = TableUser.Rows[0].Cells[8].Text.ToString();
        Label_password.Text = TableUser.Rows[0].Cells[9].Text.ToString();
    }
}
```



```

Label_payment_user.Visible = false;
Label_cat_bid.Visible = false;
Label_event_bid.Visible = false;

string str_pay;
str_pay = "SELECT * FROM Q_payment where idUser=" + iduser;
payment.SelectCommand = str_pay;
TablePayment.DataBind();

int amount_rows = TablePayment.Rows.Count;
if (amount_rows == 0)
    Label_payment_user.Visible = true;

string str_bid_cat;
str_bid_cat = "SELECT * FROM Q_cat_bid where idUser=" + iduser;
bid_cat.SelectCommand = str_bid_cat;
TableBidCat.DataBind();

amount_rows = TableBidCat.Rows.Count;
if (amount_rows == 0)
    Label_cat_bid.Visible = true;

string str_bid_event;
str_bid_event = "SELECT * FROM Q_event_bid where idUser=" + iduser;
bid_event.SelectCommand = str_bid_event;
TableBidEvent.DataBind();

amount_rows = TableBidEvent.Rows.Count;
if (amount_rows == 0)
    Label_event_bid.Visible = true;

```

## ПРИЛОЖЕНИЕ Ж

```
// поиска кота по имени и вывод всей инфы о нем
protected void Show_cat_name(object sender, EventArgs e)
{
    string name_cat = TextBox_cat.Text;
    string crit;
    crit = "SELECT * FROM Q_cat_all where nickname='" + name_cat + "'";
    DataCat.SelectCommand = crit;
    TableCat.DataBind();

    crit = "SELECT * FROM Q_cat_photo where nickname='" + name_cat + "'";
    DataPhoto.SelectCommand = crit;
    TablePhoto.DataBind();

    crit = "SELECT * FROM Q_cat_medical_card where nickname='" + name_cat + "'";
    DataMedicalCard.SelectCommand = crit;
    TableIllnesses.DataBind();

    crit = "SELECT * FROM Q_cat_vaccination_card where nickname='" + name_cat + "'";
    Data_vaccination.SelectCommand = crit;
    TableVaccination.DataBind();
}
// вывод всей инфы о котах
protected void Show_all_cat(object sender, EventArgs e)
{
    string crit;
    crit = "SELECT * FROM Q_cat_all";
    DataCat.SelectCommand = crit;
    TableCat.DataBind();

    crit = "SELECT * FROM Q_cat_photo";
    DataPhoto.SelectCommand = crit;
    TablePhoto.DataBind();

    crit = "SELECT * FROM Q_cat_medical_card";
    DataMedicalCard.SelectCommand = crit;
    TableIllnesses.DataBind();

    crit = "SELECT * FROM Q_cat_vaccination_card";
    Data_vaccination.SelectCommand = crit;
    TableVaccination.DataBind();
    Session["idcat"] = 0;
    Session["name_cat"] = "";
}
// фильтр на таблицу с медицинской инфой по состоянию
protected void Filter_state_cat(object sender, EventArgs e)
{
    string crit;
    if (List_state_cat.SelectedValue == "0")
    {
        crit = "SELECT * FROM Q_cat_medical_card";
    }
    else
    {
        crit = "SELECT * FROM Q_cat_medical_card where state_cat='" + List_state_cat.SelectedValue + "'";
    }
    DataMedicalCard.SelectCommand = crit;
    TableIllnesses.DataBind();
}
```

```

// изменение статуса кота
protected void change_status_cat(object sender, EventArgs e)
{
    string crit;
    long idcat = Convert.ToInt64(Session["idcat"]);
    if (idcat == 0)
        Err_status_cat.Visible = true;
    else
    {
        string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
        var connect = new OleDbConnection(connectionString);
        connect.Open();
        var mycom = new OleDbCommand();
        mycom.Connection = connect;
        crit = "UPDATE cat set idStatus_cat=" + List_new_status_cat.Text +
" where nickname='" + Convert.ToString(Session["name_cat"]) + "'";
        mycom.CommandText = crit;
        mycom.ExecuteNonQuery();
        connect.Close();
        crit = "Select * from Q_cat_all where nickname='" +
Convert.ToString(Session["name_cat"]) + "'";
        DataCat.SelectCommand = crit;
        TableCat.DataBind();
    }
}

// выбор медицинской записи
protected void TableIllnesses_SelectedIndexChanged(object sender, EventArgs
e)
{
    int idrow = Convert.ToInt16(TableIllnesses.SelectedIndex);
    int idcard =
Convert.ToInt16(TableIllnesses.Rows[idrow].Cells[1].Text.ToString());
    Session["IdMC"] = idcard;
    Response.Redirect("change_medical_card.aspx");
}

// выбор кота из общей таблицы и сортировка всей инфы по коту
protected void TableCat_SelectedIndexChanged(object sender, EventArgs e)
{
    int idrow = Convert.ToInt16(TableCat.SelectedIndex);
    string namecat =TableCat.Rows[idrow].Cells[2].Text.ToString();
    Session["name_cat"] =TableCat.Rows[idrow].Cells[2].Text.ToString();
    Session["idcat"] = TableCat.Rows[idrow].Cells[1].Text.ToString();
    string crit;
    crit = "SELECT * FROM Q_cat_all where nickname='" + namecat + "'";
    DataCat.SelectCommand = crit;
    TableCat.DataBind();

    crit = "SELECT * FROM Q_cat_photo where nickname='" + namecat + "'";
    DataPhoto.SelectCommand = crit;
    TablePhoto.DataBind();

    crit = "SELECT * FROM Q_cat_medical_card where nickname='" + namecat +
"'";
    DataMedicalCard.SelectCommand = crit;
    TableIllnesses.DataBind();

    crit = "SELECT * FROM Q_cat_vaccination_card where nickname='" + namecat
+ "'";
    Data_vaccination.SelectCommand = crit;
    TableVaccination.DataBind();
}

```

```

// Добавление болезни котику
protected void Add_new_ill(object sender, EventArgs e)
{
    long idcat = Convert.ToInt64(Session["idcat"]);
    if (idcat == 0)
    {
        Err_about_new_ill.Visible = true;
    }
    else
    {
        string crit = "INSERT INTO medical_card(idCat, idIllnesses,
date_of_illnesses, vet_appointment, idState_cat) " +
        "VALUES (" + idcat + ", " + List_ill.SelectedValue + ", '" +
Convert.ToString(TextBox_date_ill.Text) + "', '" + TextBox_vet_app.Text +
"', 2)";
        string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
        var connect = new OleDbConnection(connectionString);
        connect.Open();
        var mycom = new OleDbCommand();
        mycom.Connection = connect;
        mycom.CommandText = crit;
        mycom.ExecuteNonQuery();
        connect.Close();
        crit = "SELECT * FROM Q_cat_medical_card where nickname='" +
Convert.ToString(Session["name_cat"]) + "'";
        DataMedicalCard.SelectCommand = crit;
        TableIllnesses.DataBind();
        TextBox_vet_app.Text = "";
        TextBox_date_ill.Text = "";
    }
}

// создание болезни
protected void Create_ill(object sender, EventArgs e)
{
    string crit = "INSERT INTO illnesses(illnesses) Values('" +
TextBox_new_ill.Text + "')";
    string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
    var connect = new OleDbConnection(connectionString);
    connect.Open();
    var mycom = new OleDbCommand();
    mycom.Connection = connect;
    mycom.CommandText = crit;
    mycom.ExecuteNonQuery();
    connect.Close();
    Mess_about_all_good_ill.Visible = true;
    TextBox_new_ill.Text = " ";
}

// добавление прививки котику
protected void Add_new_vac(object sender, EventArgs e)
{
    long idcat = Convert.ToInt64(Session["idcat"]);
    if (idcat == 0)
    {
        Err_about_cat_vac.Visible = true;
    }
    else

```

```

        {
            if (TextBox_date_vac.Text != ""){
                string crit = "INSERT INTO vaccination_card(idCat,
idVaccination, date_vaccination) " +
                "VALUES (" + idcat + ", " + List_vac.SelectedValue + ", '" +
Convert.ToString(TextBox_date_vac.Text) + "')";
                string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;
Data Source=d:\\PIS\\DatabaseCat.accdb";
                var connect = new OleDbConnection(connectionString);
                connect.Open();
                var mycom = new OleDbCommand();
                mycom.Connection = connect;
                mycom.CommandText = crit;
                mycom.ExecuteNonQuery();
                connect.Close();
                crit = "SELECT * FROM Q_cat_vaccination_card where nickname='"
+ Convert.ToString(Session["name_cat"]) + "'";
                DataVaccination.SelectCommand = crit;
                TableVaccination.DataBind();
                TextBox_date_vac.Text = "";
            }
        }
    }

// создание прививки коту
protected void Create_vac(object sender, EventArgs e)
{
    string crit = "INSERT INTO vaccination(vaccination) Values('" +
TextBox_new_vac.Text + "')";
    string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
    var connect = new OleDbConnection(connectionString);
    connect.Open();
    var mycom = new OleDbCommand();
    mycom.Connection = connect;
    mycom.CommandText = crit;
    mycom.ExecuteNonQuery();
    connect.Close();
    Label_good_vac.Visible = true;
    TextBox_new_vac.Text = " ";
}

```

```

// фильтр на мероприятия по дате
protected void Filter_event(object sender, EventArgs e)
{
    string crit;
    if (ListTimeStatus.SelectedValue == "0")
    {
        crit = "SELECT * FROM Q_event";
    }
    else
    {
        if (ListTimeStatus.SelectedValue == "1")
        {
            crit = "SELECT * FROM Q_event where NOW() > date_event";
        }
        else
        {
            crit = "SELECT * FROM Q_event where NOW() < date_event";
        }
    }
    DataEvent.SelectCommand = crit;
    TableEvent.DataBind();
}

// добавление нового мероприятия
protected void Add_event(object sender, EventArgs e)
{
    string time_event = TextBox_time.Text + ":00";
    string crit = "INSERT INTO event(type_event, date_event, time_event,
place, activity, idVolunteer) " +
        "VALUES (" + List_type.SelectedValue + ", '" + TextBox_date.Text
+ "'", '" + time_event + "'", '" +
        "'" + TextBox_place.Text + "'", '" + TextBox_activ.Text + "'", '" +
List_volon.SelectedValue + "')";
    string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
    var connect = new OleDbConnection(connectionString);
    connect.Open();
    var mycom = new OleDbCommand();
    mycom.Connection = connect;
    mycom.CommandText = crit;
    mycom.ExecuteNonQuery();
    connect.Close();

    TableEvent.DataSourceID = "DataEvent";
    TableEvent.DataBind();

    TextBox_date.Text = "";
    TextBox_time.Text = "";
    TextBox_place.Text = "";
    TextBox_activ.Text = "";
}

```

```

// Обновление информации в мероприятии
protected void Change_event(object sender, EventArgs e)
{
    long idevent = Convert.ToInt64(Session["IdE"]);
    if (idevent == 0)
    {
        Err_event.Visible = true;
    }
    else
    {
        string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
        var connect = new OleDbConnection(connectionString);
        connect.Open();
        var mycom = new OleDbCommand();
        mycom.Connection = connect;
        string sqlQ;
        if (TextBox_date_ch.Text != "")
        {
            sqlQ = "UPDATE event SET date_event='" + TextBox_date_ch.Text +
            "" where idEvent=" + idevent;
            mycom.CommandText = sqlQ;
            mycom.ExecuteNonQuery();
        }
        if (TextBox_time_ch.Text != "")
        {
            string event_time = TextBox_time_ch.Text + ":00";
            sqlQ = "UPDATE event SET time_event='" + event_time + "" where
            idEvent=" + idevent;
            mycom.CommandText = sqlQ;
            mycom.ExecuteNonQuery();
        }
        if (TextBox_place_ch.Text != "")
        {
            sqlQ = "UPDATE event SET place='" + TextBox_place_ch.Text + ""
            where idEvent=" + idevent;
            mycom.CommandText = sqlQ;
            mycom.ExecuteNonQuery();
        }
        if (List_volont.SelectedValue != "0")
        {
            sqlQ = "UPDATE event SET idVolunteer=" +
            List_volont.SelectedValue + " where idEvent=" + idevent;
            mycom.CommandText = sqlQ;
            mycom.ExecuteNonQuery();
        }
        connect.Close();
        Response.Redirect("admin_event.aspx");
    }
}

// Выбор мероприятия
protected void TableEvent_SelectedIndexChanged(object sender, EventArgs e)
{
    int idrow = Convert.ToInt16(TableEvent.SelectedIndex);
    int idevent =
    Convert.ToInt16(TableEvent.Rows[idrow].Cells[1].Text.ToString());
    Session["IdE"] = idevent;
}

```

## ПРИЛОЖЕНИЕ И

```
// фильтры по заявкам на кота по типу и статусу
protected void Show_filter(object sender, EventArgs e)
{
    string crit;
    crit = "SELECT * FROM Q_cat_bid";
    if (ListType.SelectedValue != "0")
    {
        crit += " where type='" + ListType.SelectedItem + "'";
    }
    if (ListStatus.SelectedValue != "0")
    {
        if (ListType.SelectedValue != "0")
            crit += " and status_bid='" + ListStatus.SelectedItem + "'";
        else
            crit += "where status_bid='" + ListStatus.SelectedItem + "'";
    }
    DataBidCat.SelectCommand = crit;
    TableCatBid.DataBind();
}

// показать все заявки
protected void Show_all(object sender, EventArgs e)
{
    string crit;
    crit = "SELECT * FROM Q_cat_bid";
    DataBidCat.SelectCommand = crit;
    TableCatBid.DataBind();
}

// выбор заявки на кота
protected void TableCatBid_SelectedIndexChanged(object sender, EventArgs e)
{
    int idrow = Convert.ToInt16(TableCatBid.SelectedIndex);
    Session["IdUserBid"] =
    Convert.ToInt16(TableCatBid.Rows[idrow].Cells[2].Text.ToString());
    Session["IdBid"] =
    Convert.ToInt16(TableCatBid.Rows[idrow].Cells[1].Text.ToString());
}

// показать информацию о пользователе из заявки
protected void Show_people(object sender, EventArgs e)
{
    long iduser = Convert.ToInt64(Session["IdUserBid"]);
    if (iduser == 0)
    {
        Err_bid_show.Visible = true;
    }
    else
    {
        string crit = "Select * from all_user where idUser=" + iduser;
        DataUser.SelectCommand = crit;
        TableUser.DataBind();
        TableUser.Visible = true;
    }
}
```



```

// скрыть информацию о пользователе
protected void Not_show_people(object sender, EventArgs e)
{
    TableUser.Visible = false;
}

// изменить статус заявки и указать причину отказа
protected void ChangeStatus(object sender, EventArgs e)
{
    long iduser = Convert.ToInt64(Session["IdUserBid"]);
    if (iduser == 0)
    {
        Label_err_status.Visible = true;
    }
    else
    {
        long idbid = Convert.ToInt64(Session["IdBid"]);
        string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
        var connect = new OleDbConnection(connectionString);
        connect.Open();
        var mycom = new OleDbCommand();
        mycom.Connection = connect;
        string crit = "UPDATE bid_cat set idStatus_bid=" +
List_ch_status.SelectedValue + " where idBid=" + idbid;
        mycom.CommandText = crit;
        mycom.ExecuteNonQuery();
        if (List_ch_status.SelectedValue == "3")
        {
            if (TextBox_reason.Text != "")
            {
                crit = "UPDATE bid_cat set reason='" + TextBox_reason.Text
+ "'" where idBid=" + idbid;
                mycom.CommandText = crit;
                mycom.ExecuteNonQuery();
            }
        }
        connect.Close();
        TableCatBid.DataSourceID = "DataBidCat";
        TableCatBid.DataBind();
        Session["IdUserBid"] = 0;
    }
}

// выбор заявки на мероприятие
protected void TableEvent_SelectedIndexChanged(object sender, EventArgs e)
{
    int idrow = Convert.ToInt16(TableEvent.SelectedIndex);
    Session["IdEventBid"] =
Convert.ToInt16(TableEvent.Rows[idrow].Cells[1].Text.ToString());
}

// фильтр на заявки на мероприятие по статусу
protected void Show_event_bid(object sender, EventArgs e)
{
    string crit = "Select * from Q_admin_event where status_bid='" +
List_status_bid_event.SelectedItem+"'";
    DataEvent.SelectCommand = crit;
    TableEvent.DataBind();
}

```

```

// отображение всех мероприятий
protected void Not_show_event_bid(object sender, EventArgs e)
{
    string crit = "Select * from Q_admin_event";
    DataEvent.SelectCommand = crit;
    TableEvent.DataBind();
}
// изменение статуса заявки мероприятия
protected void Change_bid_event(object sender, EventArgs e)
{
    long idevent = Convert.ToInt64(Session["IdEventBid"]);
    if (idevent == 0)
    {
        Err_bid_event.Visible = true;
    }
    else
    {
        string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=d:\\PIS\\DatabaseCat.accdb";
        var connect = new OleDbConnection(connectionString);
        connect.Open();
        var mycom = new OleDbCommand();
        mycom.Connection = connect;
        string crit = "UPDATE bid_to_event set idStatus_bid=" +
List_ch_status_ev.SelectedValue + " where idBid=" + idevent;
        mycom.CommandText = crit;
        mycom.ExecuteNonQuery();
        connect.Close();
        TableEvent.DataSourceID = "DataEvent";
        TableEvent.DataBind();
        Session["IdEventBid"] = 0;
    }
}

```