Введение

1. Анализ предметной области
   1. Общее положение
   2. Определение проблем и актуальности разработки программного обеспечения
2. Проектирование системы
   1. Диаграмма вариантов использования
      1. Анализ актеров
      2. Диаграмма прецедентов
      3. Анализ связей расширения
      4. Модифицированная диаграмма прецедентов
   2. Диаграмма последовательности
   3. Диаграмма состояний
3. Разработка структуры базы данных
   1. Диаграмма классов
   2. ER-диаграмма
4. Разработка серверной и клиентской частей системы
   1. Диаграмма развертывания
   2. Серверная часть
   3. Клиентская часть
5. Руководство пользователя
   1. Руководство клиента портала
   2. Руководство администратора портала

Введение

В наши дни работу практически любого предприятия или организации невозможно представить себе без автоматизированных систем обработки информации. В условиях рыночной конкуренции возможность остаться на плаву получают только эффективно и рационально работающие компании. Добиться уменьшения затрат, как рабочего времени, так и денежных средств, вести оперативный учет всего что происходит в рамках предметной области компании, получать максимальную прибыть при минимальных затратах – вот лишь неполный список того, чего можно добиться внедрением в работу компании автоматизированных систем.

Немаловажным фактором является и доступность услуг, предоставляемых компанией, конечному пользователю. Чем проще, удобнее и быстрее потенциальные пользователи услуг компании могут получить к ним доступ, тем эффективнее и прибыльнее будет работа компании, тем популярнее она будет среди своих потребителей.

Целью данной работы является разработка специализированной системы, позволяющей реализовать все выше описанные факты, с точки зрения приюта для бездомных животных. Так как, такая разработка относительно не дорога и максимально ориентирована на конкретные бизнес процессы, разработка и внедрение позволит значительно уменьшить затраты компании и повысить эффективность ее работы.

Разработка системы будет производить в виде публичного интернет сайта. Такая форма АИС позволит реализовать следующие преимущества:

1. Повышения доступности услуг компании
   1. Со стационарных компьютеров
   2. С мобильных устройств
2. Обеспечения оперативности поступающей информации

В качестве платформы для развертывания системы был выбран программный пакет **Denver**. Данный пакет является свободно распространяемым и включает в себя следующие компоненты:

1. Веб-сервер Apache
2. Интерпретатор языка PHP
3. СУБД MySQL

Пакет включает в себя все необходимое для развертывания веб-портала. Так же, данный набор программных компонентов является очень распространённым среди компаний предоставляющий услуги по размещению сайтов на их серверах.

**Анализ предметной области**

**Общие положения**

Приют для бездомных животных – это место содержания бездомных животный, чаще всего кошек и собак. Приюты являются одной из важнейших составных частей защиты бездомных животных. Они обеспечивают возможность оказания оперативной помощи животным, обеспечивают облегчение страданий животных посредством ветеринарной помощи. Позволяют организовать долгосрочную заботу о животных лишившихся хозяина или еще не нашедших его.

**Определение проблем и актуальности разработки программного обеспечения**

Работа приюта самым тесным образом связана не только с самими животными, попадающими на содержание, или просто требующими заботы.

Не менее важной задачей, для любого приюта, является организация взаимодействия с обществом. Обеспечение доступности информации, которой владеет приют. Зачастую, потерянные животные так и остаются не найденными по причине отсутствия доступа к информации об их размещении в приют. Бездомные животные приюта так и не находят своих хозяев.

Все это определяет актуальность разработки АИС. Все выше описанные проблемы с легкость решаются внедрением информационной системы. Так же, АИС предоставляет массу дополнительных возможностей по развитию и улучшению работы приюта.

**Информационная система**

Информационная система – это комплекс компонент, предназначенных для обработки и хранения информации, а также других ресурсов. Она служит для организации выполнения бизнес процессов на надлежащем уровне. Каждая информационная система призвана функционировать в своей конкретной предметной области и решает задачи связанные с процессами, происходящими в этой области.

В качестве системы, разрабатываемой для приюта, была выбрана и клиент-серверная информационная система. Такая система состоит и нескольких компонент и является распределенной.

Здесь можно выделить:

серверную часть – включает в себя систему управления базами данных и дополнительное программное обеспечение, обеспечивающее функционирование системы;

клиентскую часть – включает себя приложение, доступное конечному пользователю на его компьютере или мобильном устройстве;

**Web-сайт**

Web-сайт, с точки зрения пользователя, это структурированная информация, отображаемая в виде страниц в браузере. В более широком смысле, сайт также можно определить как распределенную информационную систему, состоящую из серверной части, обеспечивающей хранение, обработку и отображение данных, и клиентской части, взаимодействующей с конечным пользователем.

Серверная часть любого сайта состоит из нескольких компонентов:

1. СУБД – отвечает за организацию контролируемого доступа к данным. Предоставляет возможности по созданию и поддержке базы данных.
2. Веб-сервер – служит для организации приема http запросов от клиентов. Каждое обращение клиента к серверу является запросом на получение ресурса – веб-страниц, файлов и т.д.
3. Препроцессор гипертекста – сценарный язык программирования, служащий для краткого описания действий, которые должна выполнять система.

Клиентская часть это набор интерактивных и статических страниц, отображающихся в браузере пользователя. Страницы (или части страниц) отображаются в результате http запросов к серверной части системы и могут быть, как сгенерированы системой, так и взяты с диска.

Страницы, формируемые препроцессором гипертекста, называются **динамическими**. Чем крупнее сайт и сложнее его структура, тем труднее при разработке обходиться без таких, динамических, страниц.

Они позволяют

отображать различные списки

отображать различные данные взятые из базы данных

обновлять блоки на сайте, не перезагружая текущей страницы

Динамические страницы повышают интерактивность сайта, позволяют оптимизировать трафик и скорость работы сайта.

Статические страницы отображаются пользователя в таком же виде, в каком хранятся на диске. Ни смотря на все более нарастающие темпы развития технологии, позволяющих создавать динамические сайты, актуальность использования статических страниц сохраняется. Золотой серединой в вопросе баланса этих двух технологий является обоснованное и продуманное использование, ориентированное на оптимальное сочетание всех характеристик сайта.

**Проектирование системы**

**Диаграмма вариантов использования**

Для понимания того что будет делать система и для того что бы правильно ее спроектировать необходимо рассмотреть функциональную модель системы.

**Выявление актеров**

Первым и одним из самых важных шагов в построении такой модели служит выявление актеров, взаимодействующих с системой.

Во время работы система постоянно взаимодействует с внешними, по отношению к ней сущностями – актерами. Актерами могут выступать как люди, так и внешние системы. Каждый актер выполняет определенную роль в системе и должен иметь имя.

Анализ актеров с точки зрения портала приюта бездомных животных показал, что роли клиентов можно обобщить в две: **хозяин животного** и **нашедший бездомное животное**. В свою очередь роль **хозяин животного** можно разбить на три дочерних роли имеющих общий признак владения животным (или намеренье им владеть): **потенциальный хозяин**, **клиент гостиницы**, **потерявший питомца**. Роли сотрудников приюта можно обобщить в две: **менеджер** и **диспетчер**.

**Выявление вариантов использования**

При взаимодействии актера с системой постоянно выполняется работа, которая образует варианты использования системы. Выявление этих вариантов и является следующим шагом на пути к построению функциональной модели системы.

Для выявления вариантов использования рассмотрим требования, которые предъявляют к системе актеры.

|  |  |
| --- | --- |
| **Потенциальный хозяин** | по требованию актера система предоставляет форму для регистрации заявки на питомца, включающую информацию об актере, необходимую для обратной связи с ним |
| **Клиент гостиницы** | по требованию актера система предоставляет форму для регистрации заявки на размещение, выдает список вольеров, свободных на указанный интервал времени и сохраняет введенную информацию в базу |
| **Хозяин потерянного питомца** | по требованию актера система выдает список животных приюта, которые удовлетворяют требованиям, заданным параметрами поиска |
| **Информатор** | по требованию актера система предоставляет интерфейс регистрации сообщения о потерянном животном и сохраняет введенные данные в базу |
| **Менеджер приюта** | по требованию актера система предоставляет форму для ввода данных о животном и сохраняет введенные данные в базу данных |
| по требованию актера система предоставляет форму для ввода информации о размещении животного в вольере питомника и сохраняет введенную информацию |
| по требованию актера предоставляется форма ввода данных о вольере питомника, введенные данные сохраняются в базу данных |
| **Диспетчер приюта** | по требованию актера система отображает список сообщений о найденных бездомных животных |
| по требованию актера система отображает список заявок на размещение питомцев в гостинице приюта |

**Детализация вариантов использования**

1. Менеджер и диспетчер получает доступ к описанным выше действиям только после прохождения им авторизации с указание логина и пароля.
2. При отображении форм для ввода информации о животном и вольере, системе необходимо загрузить справочную информацию из базы данных и отобразить ее на форме в виде выпадающего списка.
3. При выборе вольера, в который будет размещено животное, система должна показывать изображение из базы, соответствующее выдранному пункту из списка вольеров.
4. После ввода информации на любую из доступных форм системы сохранение данных должно происходить только по требованию актера.

С учетов всех замечаний диаграмма вариантов использования примет вид, приведенные на рисунке

**Реестр вариантов использования**

По результатам проведенного выше анализа, составим реестр вариантов использования.

|  |  |  |  |
| --- | --- | --- | --- |
| Код | Основной актер | Наименование | Формулировка |
| П1 | Потенциальный хозяин | Регистрация заявки на животное | Отображается список животных приюта, после выбора актером животного показывается форма для ввода информации об актере и по его требованию происходит регистрация заявки на животное приюта |
| К1 | Клиент гостиницы | Регистрация заявки на размещение | Отображается форма для выбора вольера и ввода информации и по требованию актера происходит запись введенной информации в базу данных |
| Х1 | Хозяин потерянного питомца | Поиск питомца | Отображается список животных находящихся в приюте, который по требованию актера фильтруется с учетом указанных условий |
| И1 | Информатор | Регистрация сообщения | Отображается форма для ввода данных о найденном животном и по требованию актера происходит запись введенной информации в базу данных |
| М1 | Менеджер | Добавление животного | Загружаются справочники, отображается форма ввода информации о животном и по требованию актера происходит запись в базу данных |
| М2 | Менеджер | Добавление вольера | Загружаются справочники, отображается форма ввода информации о вольере и по требованию актера происходит запись в базу данных |
| М3 | Менеджер | Размещение в вольере | Загружаются список животных и список вольеров, отображается форма для привязки животного к вольеру и дополнительной информации, по требованию актера происходит запись в базу данных |
| Д1 | Диспетчер | Просмотр списка заявок | По требованию актера отображается список заявок на размещение питомцев в гостевых вольерах питомника |
| Д2 | Диспетчер | Просмотр списка сообщений | По требованию актера отображается список сообщений о найденных бездомных животных |

**Конкретизация вариантов использования**

**П1. Регистрация заявки на животное**

Основное действующее лицо: потенциальный хозяин.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: данный вариант использования позволяет использовать систему в качестве журнала регистрации заявок на животных, размещенных в приюте на данный момент. Позволяет предоставлять пользователю только актуальные данные о находящихся в приюте животных. Результатом использования является заявка, включающая в себя следующие данные : какое животное было выбрано, информацию оставленную актером о себе, контактная информация ,комментарий.

**К1. Регистрация заявки на размещение**

Основное действующее лицо: клиент гостиницы.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: вариант использования предусматривает наличие в системе интерфейса выбора свободного вольера. Здесь система выступает в качестве упрощенной гостевой книги. Актеру предоставляется возможность выбрать промежуток времени, в днях, на который он хотел бы поместить своего питомца. В зависимости от выбранного промежутка дат, актеру предлагается выбрать один из свободных вольеров. Результатом использования является заявка на размещение животного в гостиничном вольере питомника, включающая всю оставленную актером информацию и дату ее регистрации.

**Х1. Поиск питомца**

Основное действующее лицо: хозяин потерянного питомца.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: в данном варианте использования система выступает в качестве поискового инструмента. Системе предоставляет интерактивный полнотекстовый поиск по описанию каждого из питомцев, содержащихся в питомнике. Результатом использования являет список животных с выводом дополнительной информации о них. В результирующий список попадают только животные отвечающие критериям поиска, остальные – скрываются.

**И1. Регистрация сообщения**

Основное действующее лицо: информатор.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: в данном варианте использования система выступает в качестве сервиса приема сообщений о бездомных животных, найденных на улице. Актеру предоставляется интерфейс интерактивной карты, позволяющий указать точные координаты места, где он столкнулся с животным. Так же, предоставляется возможность загрузить фотографию с изображением животного. Результатом использования будет регистрация в системе сообщения для диспетчера приюта. Сообщения отображаются в отдельном интерфейсе, доступном только авторизованным пользователям сайта.

**М1. Добавление животного**

Основное действующее лицо: менеджер.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: система выступает в качестве редактора, позволяющего добавлять в базу животных. Результатом использования является запись базы, включающая в себя формацию о типе животного, его фотографию и основные сведенья. Система предоставляет удобный интерфейс по указанию всех этих параметров и загрузке изображения на сервер.

**М2. Добавление вольера**

Основное действующее лицо: менеджер.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: вариант использования предусматривает использование системы в качестве редактора справочника вольеров. Актеру предоставляется возможность загрузить фотографию вольера и указать необходимую информацию. Результатом использования будет запись базы питомника, включающая информацию о вольере.

**М3. Размещение в вольере**

Основное действующее лицо: менеджер.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: прежде чем животное, добавленное в базу питомника, будет отображено в публичном списке, доступном пользователям, актер вносит данные о его размещении в одном из вольеров. Также указывается срок, в виде интервала дат, на который животное помещается в вольер. В случае размещения животного в один из вольеров питомника достаточно указать только дату размещения. При размещении в гостевой вольер указание даты окончания размещения – обязательно. Результатом использования системы будет связка животного и вольера, действительная определенный промежуток времени.

**Д1. Просмотр списка заявок**

Основное действующее лицо: диспетчер.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: вариант использования предусматривает наличие в системе оперативного отчета. Отчет должен отображать список заявок на размещение питомцев в гостевые вольеры. Каждая строка отчета соответствует одной заявке и включает в себя всю информацию, сохраненную в базе. Результатом использования является набор строк, отображаемых в виде таблицы с возможностью оперативного обновления.

**Д2. Просмотр списка сообщений**

Основное действующее лицо: диспетчер.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: вариант использования предусматривает наличие в системе оперативного отчета. Отчет должен отображать список сообщений о найденных животных. Каждая строка отчета соответствует одному сообщению и включает в себя всю информацию, сохраненную в базе. Результатом использования является набор строк, отображаемых в виде таблицы с возможностью оперативного обновления.

**Разработка технического задания**

**Общее описание системы**

**Наименование**

Полное наименование системы: web-сайт приюта для бездомных животных «Задний дворик».

Условное обозначение: web-сайт «Задний дворик»

**Номер договора**

Договор №100007 от 1 января 2015 года на разработку, внедрение и сопровождение веб-сайта, служащего для автоматизации работы приюта для бездомных животных.

**Наименования Разработчика**

Разработчик: Закрытое акционерное общество «MerlinWEB»

Адрес: 430005, Саранск, ул. Московская, д.62

Тел.: (8342) 38-80-20, факс: (8342) 43-56-58

Банковские реквизиты: ЗАО «MerlinWEB», ИНН 5702001741, р/сч № 50603411400020007730 в АКБ Сбербанк России, БИК 044575580, корр. счет № 40101810400000003267

**Наименования Заказчика**

Заказчик: Открытое акционерное общество «Задний дворик»

Адрес: 430005, Саранск, ул. Ленина, д.12

Тел.: (8342)32-75-35

Банковские реквизиты: ОАО «Задний дворик», ИНН 4511007339, р/сч № 10603410900020004054 в АКБ Сбербанк России, БИК 044579737, корр. счет № 30101820400000000418

**Основание для проведения работ**

Основанием для проведения работ по созданию web-cайта «Задний дворик» являются следующие документы:

* Договор № 20 от 01.01.2015

**Сроки начала и окончания работ**

Дата начала работ: 15.01.2015

Дата окончания работ: 01.03.2015

**Источники и порядок финансирования работ**

Финансирование работ осуществляется из средств ОАО «Задний дворик». Порядок финансирования работ определяется условиями Договора № 20 от 01.01.2015

**Порядок оформления и предъявления Заказчику результатов работ**

Работы по созданию Системы производятся и принимаются поэтапно.

По окончании каждого из этапов работ Разработчик представляет Заказчику соответствующую документацию и подписанный со стороны Разработчика Акт сдачи-приемки работ.

**Список действующих лиц**

Посетитель сайта – доступна регистрация заявки на животное, на размещение питомца в гостинице; регистрация сообщения о найденном животном; поиск пропавшего питомца;

Менеджер приюта – доступно редактирование списка животных, вольеров; размещение животного в вольер;

Диспетчер приюта – доступен просмотр списка сообщений о найденных животный, списка заявок на размещение в гостинице;

**Структура проектируемой системы**

**Назначение системы**

Web-сайт «Задний дворик» - система предназначенная для автоматизации следующий процессов:

1. Регистрации заявок от клиентов приюта
2. Сбора информации о бездомных животных
3. Ведения учета бездомных животных приюта
4. Доступа клиентов приюта к информации о животных приюта

**Цели создания системы**

1. Создание единого механизма автоматизации процессов компании
2. Создание единой базы данных по животным питомника
3. Публикация информации в сети интернет, для потенциальных клиентов приюта

**Характеристика объекта автоматизации**

Объектом автоматизации является набор процессов, которые имеют место в рамках осуществления работы с клиентами и питомцами приюта.

В приложении web-сайт «Задний Дворик» предусмотрена возможность построения интерактивных отчетов:

* Список заявок на размещение питомца в гостинице;
* Список сообщений о найденных бездомных животных.
  + 1. **Требования для развертывания системы**

**Применяемые стандарты**

Система должна соответствовать всем стандартам интерфейса пользователя Microsoft Windows.

**Требования к среде выполнения**

Минимальные системные требования для сервера:

* 2 Гб оперативной памяти;
* 100 Gb свободного дискового пространства;
* процессор с тактовой частотой 2500 MHz;
* Операционная система Microsoft Windows Server 2008.

Минимальные системные требования для клиентской части:

* 500 Mb оператвной памяти;
* 10 Gb свободного дискового пространства;
* процессор с тактовой частотой 1500 MHz;
* Операционная система Microsoft Windows Server 2008.

**Требования к СУБД и доступу к данным.**

В ядре системы должна быть представлена промышленная СУБД реляционного доступа.

**Прочие требования**

**Требования к структуре системы**

Web-сайт «Задний дворик» предназначена для автоматизации работы приюта бездомных животных. Автоматизации подлежат операции:

регистрации животных на портале

размещения животных в вольерах

регистрации заявок от клиентов

формирования отчетной информации

предоставления информации клиентам портала

Функциональная структура Системы должна включать основные прикладные подсистемы, выполняющие задачи автоматизации обмена информацией и обработки запросов клиентов и сотрудников портала, для корректной консультации клиентов, а также для предоставления информационных услуг.

**Требования к применимости**

Интерфейс Web-сайта «Задний дворик» для клиентов и сотрудников должен обладать свойствами удобства и интуитивной ясности и не требовать дополнительной подготовки пользователей.

**Требования к надежности**

Web-сайт «Задний дворик» должен быть доступен круглосуточно.

Время, затрачиваемое на обслуживание системы не должно превышать 3% от общего времени работы.

Среднее время безотказной работы – 10 рабочих дней. Максимальная норма ошибок или дефектов – 1 ошибка на десять тысяч строк кода.

**Требования к производительности**

Система должна быть способна поддерживать минимум 100 одновременно работающих пользователей, связанных с общей базой данных.

Время отклика для типичных задач – не более 2 секунд, для сложных задач – не более 5 секунд.

**Требования к пригодности к эксплуатации**

Система должна быть способна поддерживать минимум 100 одновременно работающих пользователей, связанных с общей базой данных и иметь возможность увеличить их количество на случай увеличения числа посетителей сайта.

Пользователи АСУ должны иметь базовые навыки работы с операционными системами Microsoft (Microsoft Windows 7, 8, 8.1).

Техническое обслуживание и администрирование оборудования АСУ должно выполняться специалистами, имеющими соответствующую квалификацию и навыки выполнения работ.

Все администраторы АСУ должны иметь квалификацию «инженер» и обязательные навыки администрирования сети на основе операционной системы Microsoft Windows 7.

Обновление версий должно осуществляться в автоматизированном режиме на основе системы контроля версий и системы (сервера) обновления версий на рабочих местах пользователей.

**Требования к способам и средствам связи для информационного обмена между компонентами системы**

Информационный обмен между подсистемами должен осуществляться через единое информационное пространство и посредством использования стандартизированных протоколов и форматов обмена данными.

Все компоненты подсистем АСУ должны функционировать в пределах единого логического пространства, обеспеченного интегрированными средствами серверов данных и серверов приложений.

**Требования к совместимости со смежными системами**

Программное обеспечение системы должно обеспечивать интеграцию и совместимость на информационном уровне с другими системами. Информационная совместимость должна обеспечивается, на уровне экспорта-импорта XML-документов.

Требования к составу данных и режимам информационного обмена между подсистемами АСУ и системами, эксплуатирующимся на объекте автоматизации, определяются в общем регламенте взаимодействия.

**Требования к перспективам развития системы**

АСУ должна:

* иметь длительный жизненный цикл,
* быть построена с использованием стандартизованных и эффективно сопровождаемых решений,
* быть реализована как открытая система, и должна допускать наращивание функциональных возможностей,
* обеспечивать возможность модернизации как путем замены технического и общего программного обеспечения (ПО), так и путем совершенствования информационного обеспечения.

**Требования к показателям назначения**

Целевое назначение системы должно сохраняться на протяжении всего срока эксплуатации web-сайта «Задний дворик». Срок эксплуатации определяется сроком устойчивой работы аппаратных средств вычислительных комплексов, своевременным проведением работ по замене (обновлению) аппаратных средств, по сопровождению программного обеспечения системы и его модернизации. Время выполнения запросов информации в АСУ определяется на стадии проектирования системы.

Специальные требования к вероятностно-временным характеристикам, при которых сохраняется целевое назначение web-сайта «Задний дворик», определяются соответствующими требованиями к прикладным системам.

Прочие показатели назначения АСУ разрабатываются после проведения пред-проектного обследования.

**Построение дополнительных диаграмм**

**Диаграмма последовательности**

Диаграмма последовательности служит для упорядочивания взаимодействий объектов системы по времени. На ней отображаются сигналы и сообщения, которыми обмениваются объекты системы. Основными элементами диаграммы последовательности являются объекты, изображаемые в виде прямоугольников. Каждый из объектов имеет вертикальную пунктирную линию – «линию жизни». Для отображения периодов активности объекта используется прямоугольное выделение на «линии жизни» объекта. Сообщения провоцирующие ту или иную активность объекта так же отображаются на диаграмме, в виде переходом между «линиями жизни» объектов.

Рассмотрим диаграмму последовательности на примере варианта использования системы «М3. Размещение в вольере».

В данном случае, в качестве объекта, инициирующего взаимодействие, выступает менеджер приюта. Во взаимодействии принимают участие две формы, сервер и СУБД. Диаграмма наглядно иллюстрирует какие объекты задействуются во взаимодействии, инициированном менеджером при размещении животного в вольер.

**Построение диаграммы состояний**

Диаграмма состояний показывает все возможные состояния системы, в которые она может переходить под влиянием внешних факторов. Основными элементами диаграммы состояний являются «Состояние» и «Переход». Факт смены одного состояния другим изображается с помощью перехода. Переход осуществляется при наступлении некоторого события: окончания выполнения деятельности, получении объектом сообщения или приемом сигнала. Если при срабатывании перехода возможно ветвление, в имени перехода используется сторожевое условие.

Рассмотрим диаграмму состояний на примере варианта использования системы «М3. Размещение в вольере».

Приеденная диаграмма конкретизирует состояния, в которые может перейти система. Она имеет начальную и конечную точки. Так же, на диаграмме присутствует ветвление, обусловленное наличием или отсутствием свободных вольеров в питомнике. Диаграммой предусмотрен возврат к вводу данных в случае не возможности дальнейшего продвижения к конечной точке.

**Построение диаграммы классов**

На диаграммах классов показываются классы, интерфейсы и отношения между ними. Класс – это основной строительный блок проектируемой системы. Каждый класс имеет название, атрибуты и операции и показывается на диаграмме в виде прямоугольника, разделенного на три области. В верхней области указывается название класса, далее названия атрибутов – во второй и названия методов – в третьей.

На диаграммах классов обычно показывают ассоциации и обобщения. Каждая ассоциация несет информацию о связях между объектами внутри проектируемой системы. Обобщение на диаграммах классов используется, чтобы показать связь между классом-родителем и классом-потомком.

**Проектирование базы данных**

**Планирование развертывания системы**

Для проектируемой системы автоматизации была выбрана распределенная клиент-серверная архитектура. Для такой архитектуры характерно наличие сервера и клиентов, имеющих доступ к серверу. Таким образом для web-сайта «Задний дворик» мы имеем диаграмму развертывания изображённую на рисунке.

**Выбор средств разработки для проектируемой системы**

**Выбор средств и инструментов для разработки базы данных**

Разработкой, распространением и поддержкой MySQL, наиболее популярной системы управления базами данных (СУБД) с открытым исходным кодом, занимается компанией MySQL AB.

MySQL AB - коммерческая компания, основанная разработчиками MySQL, строит свой бизнес на предоставлении услуг, так или иначе связанных с СУБД MySQL. Кроме того, на сайте MySQL (http://www.mysql.com/) представлена наиболее актуальная информация о СУБД MySQL.

MySQL — это система управления базами данных.

База данных представляет собой структурированный набор данных. Она может содержать различную информацию - от простого списка покупок до огромного объема данных, используемого в корпоративной сети.

MySQL - это система управления реляционными базами данных.

Реляционная база данных хранит информацию в отдельных таблицах, а не в одном большом хранилище, благодаря чему достигается высокая производительность и гибкость. Часть "SQL" слова "MySQL" обозначает "Structured QueryLanguage" ("Язык структурированных запросов"). SQL - наиболее общий стандартизованный язык доступа к базам данных; он соответствует стандарту ANSI/ISO SQL. Стандарт SQL впервые был принят в 1986 году и на настоящее время существует несколько его версий. В настоящем руководстве "SQL-92" ссылается на стандарт, принятый в 1992 году, "SQL:1999" - на стандарт, принятый в 1999 году, и "SQL:2003" - на текущую версию стандарта. В дальнейшем под "стандартом SQL" имеется в виду текущая версия данного стандарта.

MySQL - это система с открытым исходным кодом.

Открытость исходного кода означает, что любой желающий имеет возможность использовать и модифицировать это программное обеспечение по своему усмотрению. Получить и развернуть программное обеспечение MySQL можно из Internet, причем совершенно бесплатно. Каждый пользователь, при желании, может изучить исходные тексты и изменить их в соответствии со своими потребностями. Программное обеспечение MySQL распространяется по лицензии GPL (GNU General Public License), которая регламентирует, что разрешено, а что нет в отношении программного обеспечения.

Сервер баз данных MySQL - очень быстрый, надежный и простой в эксплуатации.

Если это как раз то, что вы ищете, стоит с ним поработать. Сервер MySQL включает в себя практичный набор средств, разработанных в тесной кооперации с сообществом пользователей. Изначально сервер MySQL был разработан для более быстрого управления большими базами данных, чем существующие решения в этой области, и на протяжении ряда лет успешно эксплуатировался в средах, к которым предъявлялись весьма высокие требования. Несмотря на то, что MySQL пребывает в непрекращающемся процессе разработки, на сегодняшний день он предоставляет богатый набор удобных в эксплуатации средств и функций. Присущие серверу MySQL возможности сетевого взаимодействия, производительность и безопасность делают его удачным вариантом для работы с базами данных в Internet.

Сервер MySQL работает в клиент-серверных и встроенных системах.

СУБД MySQL является клиент-серверной системой, включающей многопоточный SQL-сервер, поддерживающий различные платформы, несколько клиентских программ и библиотек, инструменты администрирования и широкий диапазон программных интерфейсов приложений (API-интерфейсов).

Сервер MySQL существует также и в форме встраиваемой многопоточной библиотеки, которую можно связывать с разрабатываемыми приложениями, чтобы получить более компактные, быстрые и легкоуправляемые продукты.

Доступен огромный объем программного обеспечения MySQL, написанного независимыми разработчиками.

**Выбор средств и инструментов для разработки программы**

**Язык написания сценариев PHP**

PHP – это широко используемый язык сценариев общего назначения с открытым исходным кодом. Аббревиатура PHP означает “Hypertext Preprocessor (Препроцессор Гипертекста)". Синтаксис языка берет начало из C, Java и Perl. PHP достаточно прост для изучения. Преимуществом PHP является предоставление web-разработчикам возможности быстрого создания динамически генерируемых web-страниц. Подробнее о преимуществах PHP можно узнать здесь. Важным преимуществом языка PHP перед такими языками, как языков Perl и C заключается в возможности создания HTML документов с внедренными командами PHP. Подробнее об этой возможность смотрите здесь.

Значительным отличием PHP от какого-либо кода, выполняющегося на стороне клиента, например, JavaScript, является то, что PHP-скрипты выполняются на стороне сервера. Вы даже можете сконфигурировать свой сервер таким образом, чтобы HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта.

PHP позволяет создавать качественные Web-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем.

PHP прост для освоения, и вместе с тем способен удовлетворить запросы профессиональных программистов. Даже если Вы впервые услышали о PHP, изучить этот язык не составит для Вас большого труда. Мы не сомневаемся, что изучив основы PHP в течение нескольких часов, вы уже сможете создавать простые PHP-скрипты. Язык PHP постоянно совершенствуется, и ему наверняка обеспечено долгое доминирование в области языков web -программирования, по крайней мере, в ближайшее время.

Главным фактором языка РНР является практичность. РНР должен предоставить программисту средства для быстрого и эффективного решения поставленных задач. Практический характер РНР обусловлен пятью важными характеристиками:

* традиционностью;
* простотой;
* эффективностью;
* безопасностью;
* гибкостью.

Существует еще одна «характеристика», которая делает РНР особенно привлекательным: он распространяется бесплатно! Причем, с открытыми исходными кодами ( Open Source ).

**Традиционность**

Язык РНР будет казаться знакомым программистам, работающим в разных областях. Многие конструкции языка позаимствованы из Си, Perl.

Код РНР очень похож на тот, который встречается в типичных программах на С или Pascal. Это заметно снижает начальные усилия при изучении РНР. PHP — язык, сочетающий достоинства Perl и Си и специально нацеленный на работу в Интернете, язык с универсальным (правда, за некоторыми оговорками) и ясным синтаксисом.

И хотя PHP является довольно молодым языком, он обрел такую популярность среди web-программистов, что на данный момент является чуть ли не самым популярным языком для создания web-приложений (скриптов).

**Простота**

Сценарий РНР может состоять из 10 000 строк или из одной строки — все зависит от специфики вашей задачи. Вам не придется подгружать библиотеки, указывать специальные параметры компиляции или что-нибудь в этом роде. Механизм РНР просто начинает выполнять код после первой экранирующей последовательности (<?) и продолжает выполнение до того момента, когда он встретит парную экранирующую последовательность (?>). Если код имеет правильный синтаксис, он исполняется в точности так, как указал программист.

PHP — язык, который может быть встроен непосредственно в html -код страниц, которые, в свою очередь будут корректно обрабатываться PHP -интерпретатором. Мы можем использовать PHP для написания CGI-сценариев и избавиться от множества неудобных операторов вывода текста. Мы можем привлекать PHP для формирования HTML-документов, избавившись от множества вызовов внешних сценариев.

Большое разнообразие функций PHP избавят вас от написания многострочных пользовательских функций на C или Pascal .

**Эффективность**

Эффективность является исключительно важным фактором при программировании для многопользовательских сред, к числу которых относится и web .

Очень важное преимущество PHP заключается в его «движке». «Движок» PHP не является ни компилятором, ни интерпретатором. Он является транслирующим интерпретатором. Такое устройство «движка» PHP позволяет обрабатывать сценарии с достаточно высокой скоростью.

По некоторым оценкам, большинство PHP-сценариев (особенно не очень больших размеров) обрабатываются быстрее аналогичных им программ, написанных на Perl. Однако, чтобы не делали разработчики PHP, откомпилированные исполняемые файлы будут работать значительно быстрее – в десятки, а иногда и в сотни раз. Но производительность PHP вполне достаточна для создания вполне серьезных web-приложений. Подробно об устройстве и характеристиках «движка» PHP можно ознакомиться здесь.

**Безопасность**

РНР предоставляет в распоряжение разработчиков и администраторов гибкие и эффективные средства безопасности, которые условно делятся на две категории: средства системного уровня и средства уровня приложения.

1. Средства безопасности системного уровня

В РНР реализованы механизмы безопасности, находящиеся под управлением администраторов; при правильной настройке РНР это обеспечивает максимальную свободу действий и безопасность. РНР может работать в так называемом безопасном режиме (safe mode), который ограничивает возможности применения РНР пользователями по ряду важных показателей. Например, можно ограничить максимальное время выполнения и использование памяти (неконтролируемый расход памяти отрицательно влияет на быстродействие сервера). По аналогии с cgi-bin администратор также может устанавливать ограничения на каталоги, в которых пользователь может просматривать и исполнять сценарии РНР, а также использовать сценарии РНР для просмотра конфиденциальной информации на сервере (например, файла passwd).

2. Средства безопасности уровня приложения

В стандартный набор функций РНР входит ряд надежных механизмов шифрования. РНР также совместим с многими приложениями независимых фирм, что позволяет легко интегрировать его с защищенными технологиями электронной коммерции (e-commerce). Другое преимущество заключается в том, что исходный текст сценариев РНР нельзя просмотреть в браузере, поскольку сценарий компилируется до его отправки по запросу пользователя. Реализация РНР на стороне сервера предотвращает похищение нетривиальных сценариев пользователями, знаний которых хватает хотя бы для выполнения команды View Source.

Подробно о безопасности PHP можно ознакомиться здесь

**Гибкость**

Поскольку РНР является встраиваемым (embedded) языком, он отличается исключительной гибкостью по отношению к потребностям разработчика. Хотя РНР обычно рекомендуется использовать в сочетании с HTML, он с таким же успехом интегрируется и в JavaScript, WML, XML и другие языки. Кроме того, хорошо структурированные приложения РНР легко расширяются по мере необходимости (впрочем, это относится ко всем основным языкам программирования).

Нет проблем и с зависимостью от браузеров, поскольку перед отправкой клиенту сценарии РНР полностью компилируются на стороне сервера. В сущности, сценарии РНР могут передаваться любым устройствам с браузерами, включая сотовые телефоны, электронные записные книжки, пейджеры и портативные компьютеры, не говоря уже о традиционных ПК. Программисты, занимающиеся вспомогательными утилитами, могут запускать РНР в режиме командной строки.

Поскольку РНР не содержит кода, ориентированного на конкретный web-сервер, пользователи не ограничиваются определенными серверами (возможно, незнакомыми для них). Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold и Zeus — РНР работает на всех перечисленных серверах. Поскольку эти серверы работают на разных платформах, РНР в целом является платформенно-независимым языком и существует на таких платформах, как UNIX, Solaris, FreeBSD и Windows 95/98/NT/2000/XP/2003.

Наконец, средства РНР позволяют программисту работать с внешними компонентами, такими как Enterprise Java Beans или СОМ-объекты Win32. Благодаря этим новым возможностям РНР занимает достойное место среди современных технологий и обеспечивает масштабирование проектов до необходимых пределов.

**Бесплатное распространение**

Стратегия Open Source, и распространение исходных текстов программ в массах, оказало несомненно благотворное влияние на многие проекты, в первую очередь — Linux, хотя и успех проекта Apache сильно подкрепил позиции сторонников Open Source. Сказанное относится и к истории создания РНР, поскольку поддержка пользователей со всего мира оказалась очень важным фактором в развитии проекта РНР.

Принятие стратегии Open Source и бесплатное распространение исходных текстов РНР оказало неоценимую услугу пользователям. Вдобавок, отзывчивое сообщество пользователей РНР является своего рода «коллективной службой поддержки», и в популярных электронных конференциях можно найти ответы даже на самые сложные вопросы.

**Описание интерфейса**

Интерфейс доступный клиенту приюта исполнен в виде интернет сайта, доступного через браузер на компьютере клиента. После открытия web-сайта пользователю отображается стартовая страница.

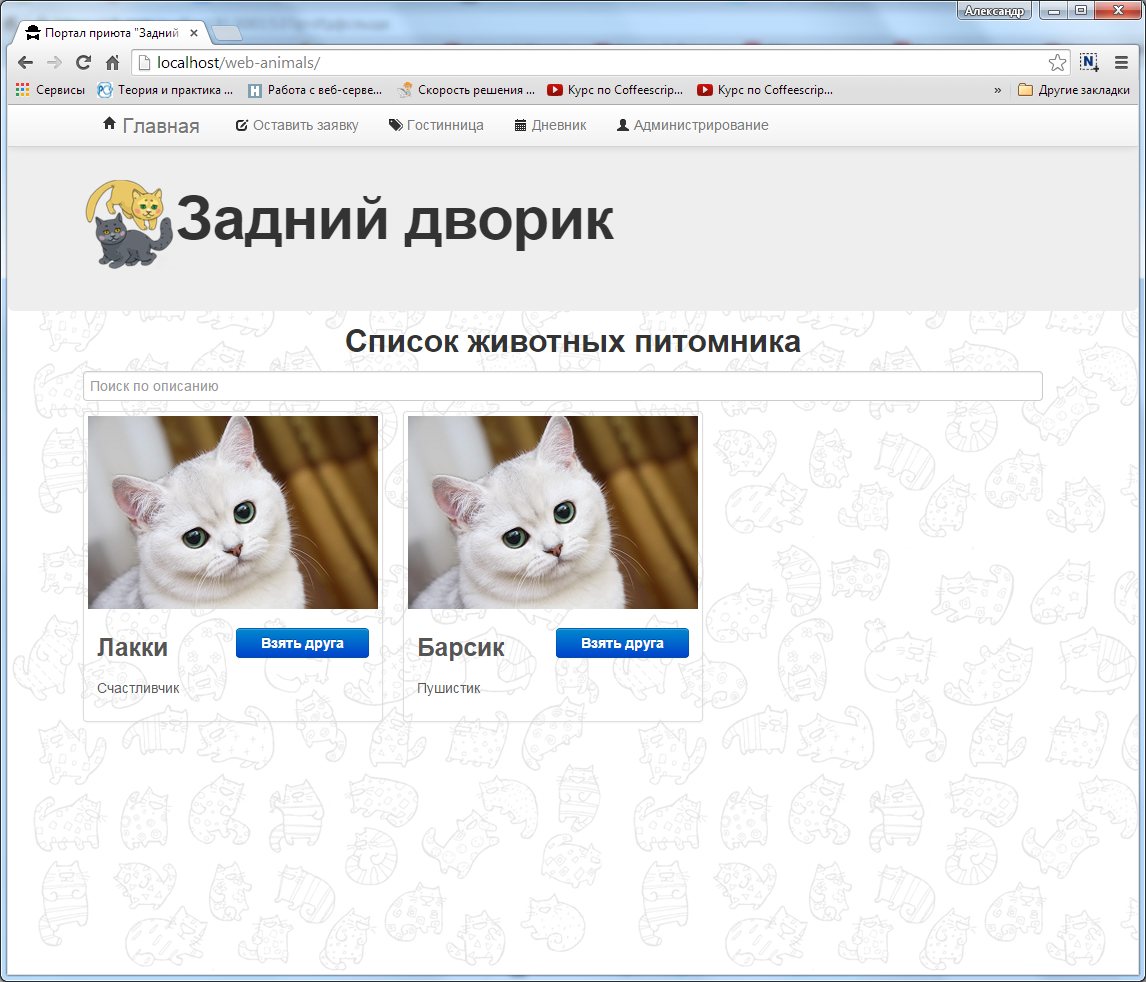
**Стартовая страница**

Стартовая страница сайта состоит из следующих функциональных элементов:

шапка сайта – содержит название сайта и логотип;

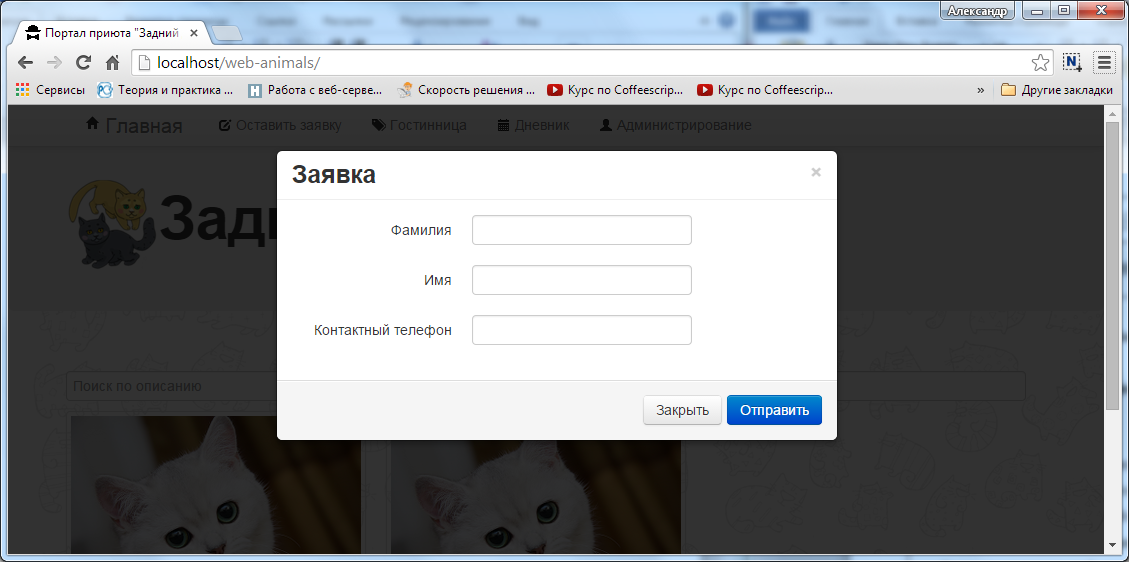
главное меню – переходы, доступные пользователю сайта;

блок контента – блок, содержащий информационное наполнение страницы.

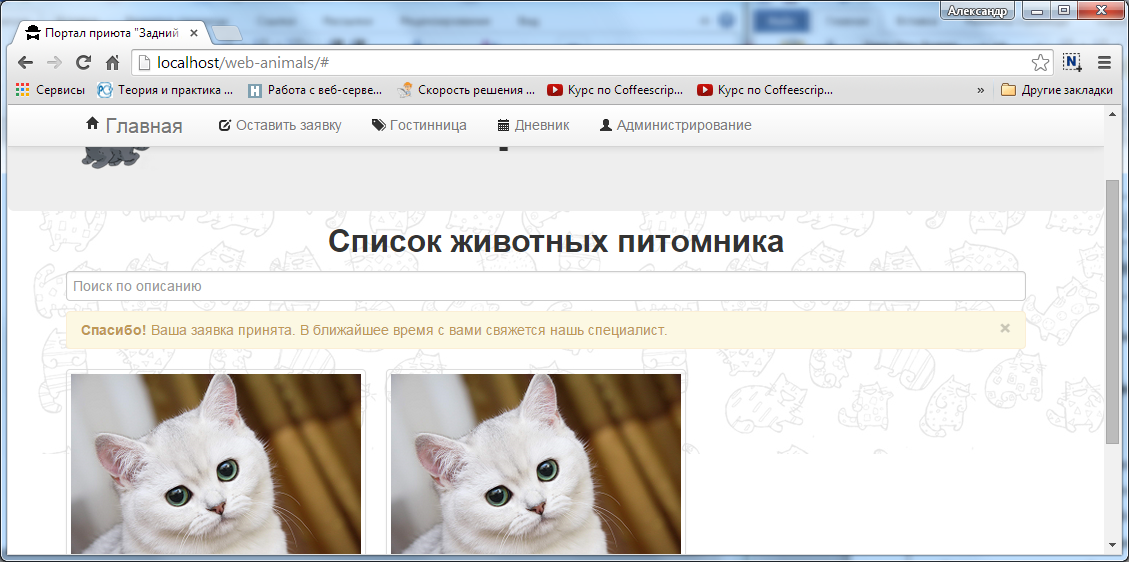


Блок контента стартовой страницы содержит список животных питомника, а так же строку поиска по этому списку. Поиск осуществляется автоматически, по мере ввода текста в поле.

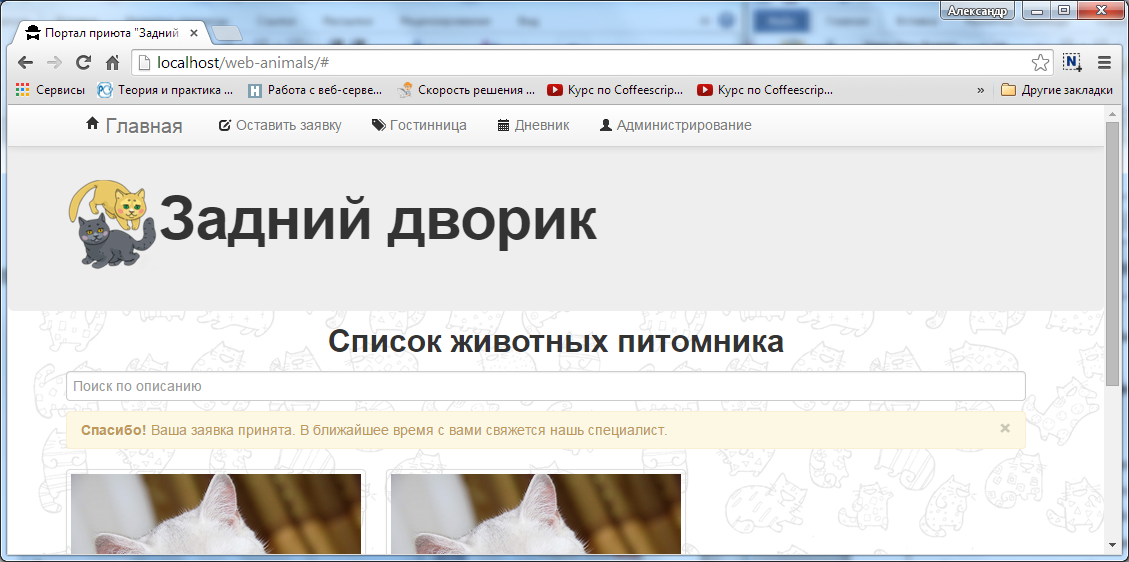
Пользователю доступна кнопка «Взять друга», нажатие на которую приводит к появлению формы регистрации заявки на выбранное животное.



Форма состоит из трех полей ввода – «Фамилия», «Имя», «Контактный телефон». При нажатии на кнопку «Отправить» происходит отправка формы на сервер. Пользователь, при этом получает уведомление о том, что его заявка принята.



**Главное меню**



Главное меню сайта выполнено в виде горизонтально расположенного списка и состоит из следующих пунктов:

«Главная» – переход на главную (стартовую страницу) сайта;

«Оставить заявку» - переход на страницу регистрации заявки о найденном бездомном животном;

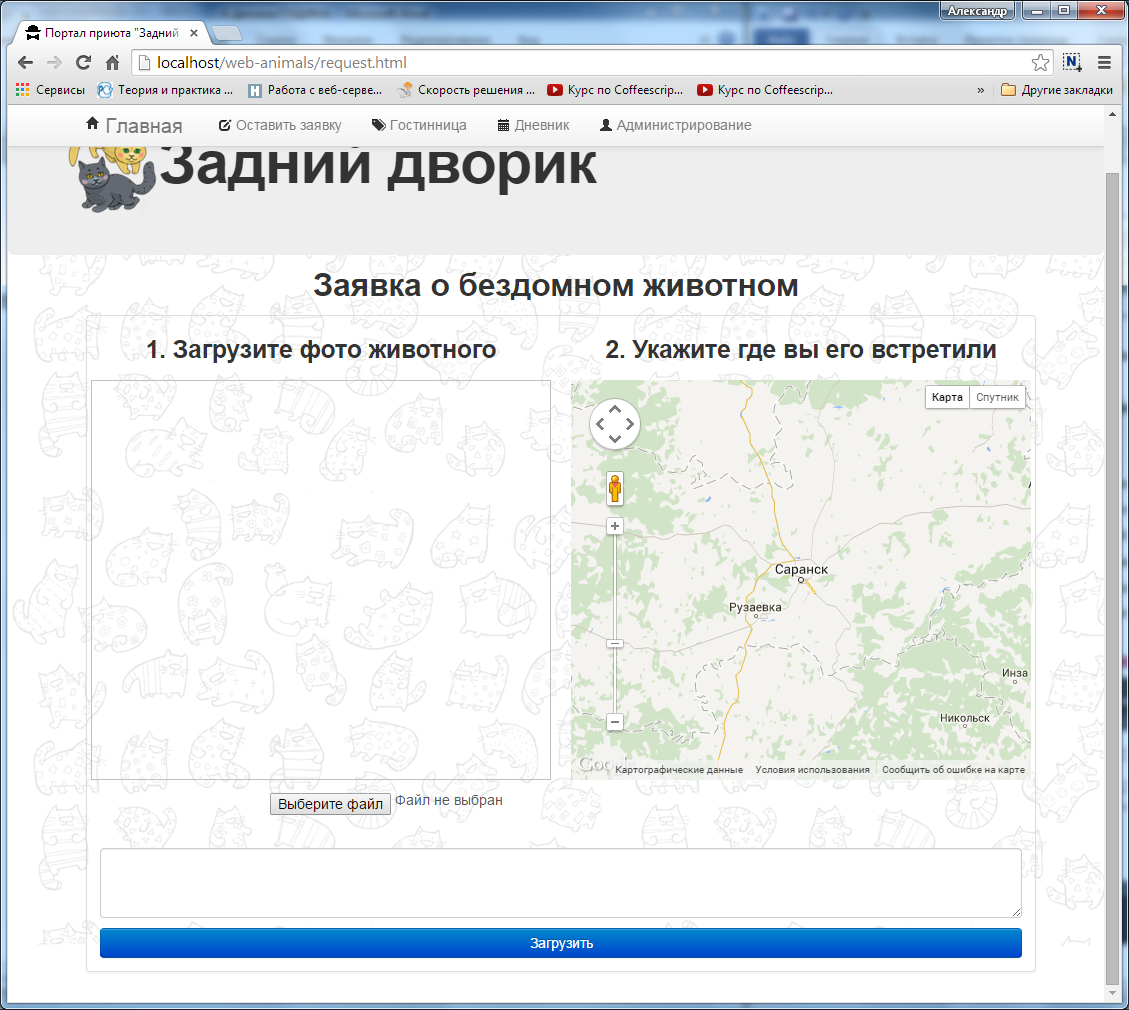
«Гостиница» - переход на страницу регистрации заявки на размещение животного в гостевом вольере приюта;

«Дневник» - переход на страницу с новостями приюта

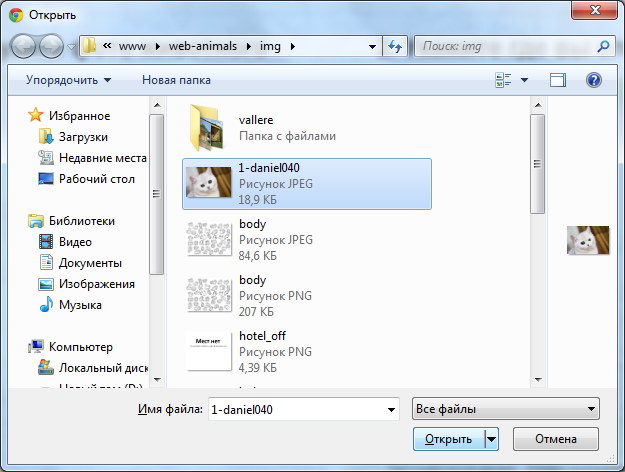
«Администрирование» - переход на форму авторизации (после успешной авторизации пользователю доступен интерфейс администратора портала)

**Страница регистрации заявки о найденном бездомном животном**

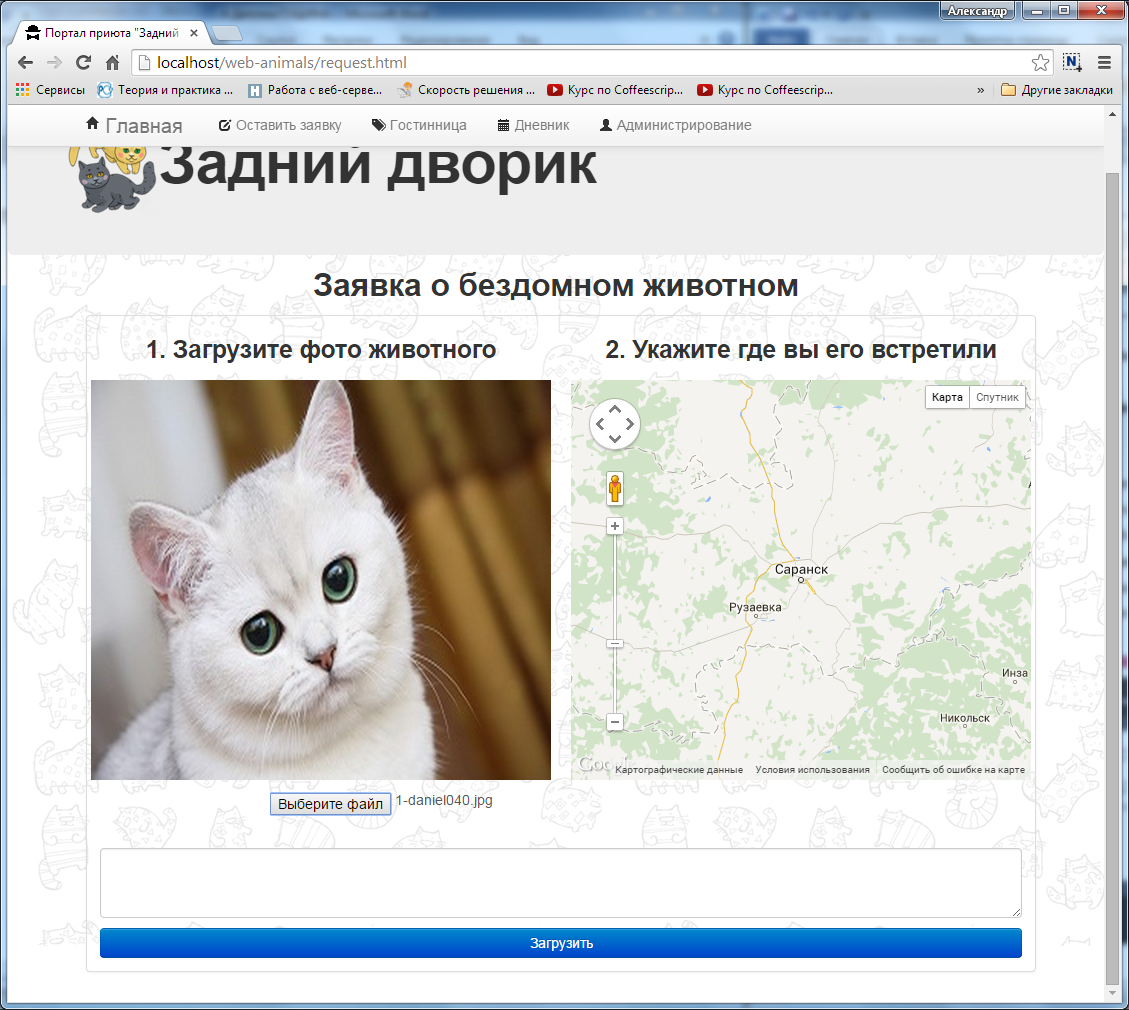
При переходе в пункт меню «Оставить заявку» в блоке контента отображается форма для регистрации заявки о бездомном животном.



Пользователю доступна возможность загрузить фотографию. После нажатия кнопки «Выберите файл» открывается стандартный диалог выбора файла.

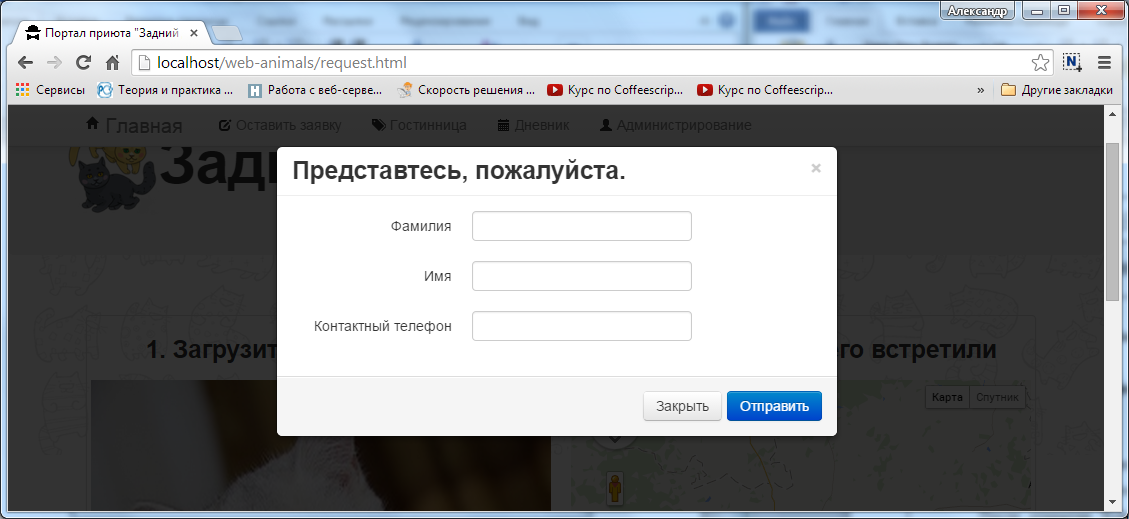


После окончания загрузки выбранного файла в блоке над кнопкой выбора будет отображено выбранное фото.



Так же, пользователю доступна интерактивная карта. С помощью мышки, перетаскивая карту, необходимо указать место, где было обнаружено бездомное животное. При регистрации заявки в базу приюта запишутся GPS координаты центра карты.

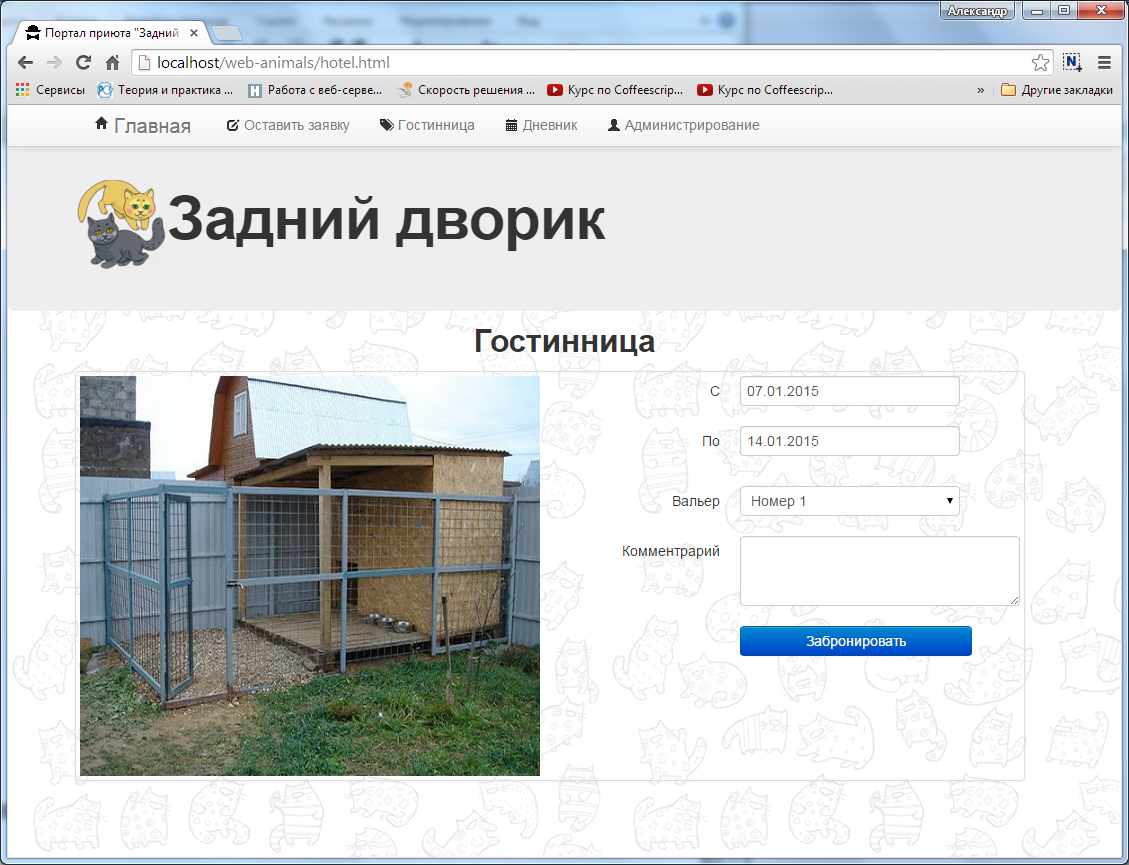
После нажатия кнопки загрузить всплывает форма ввода личной информации о пользователе.



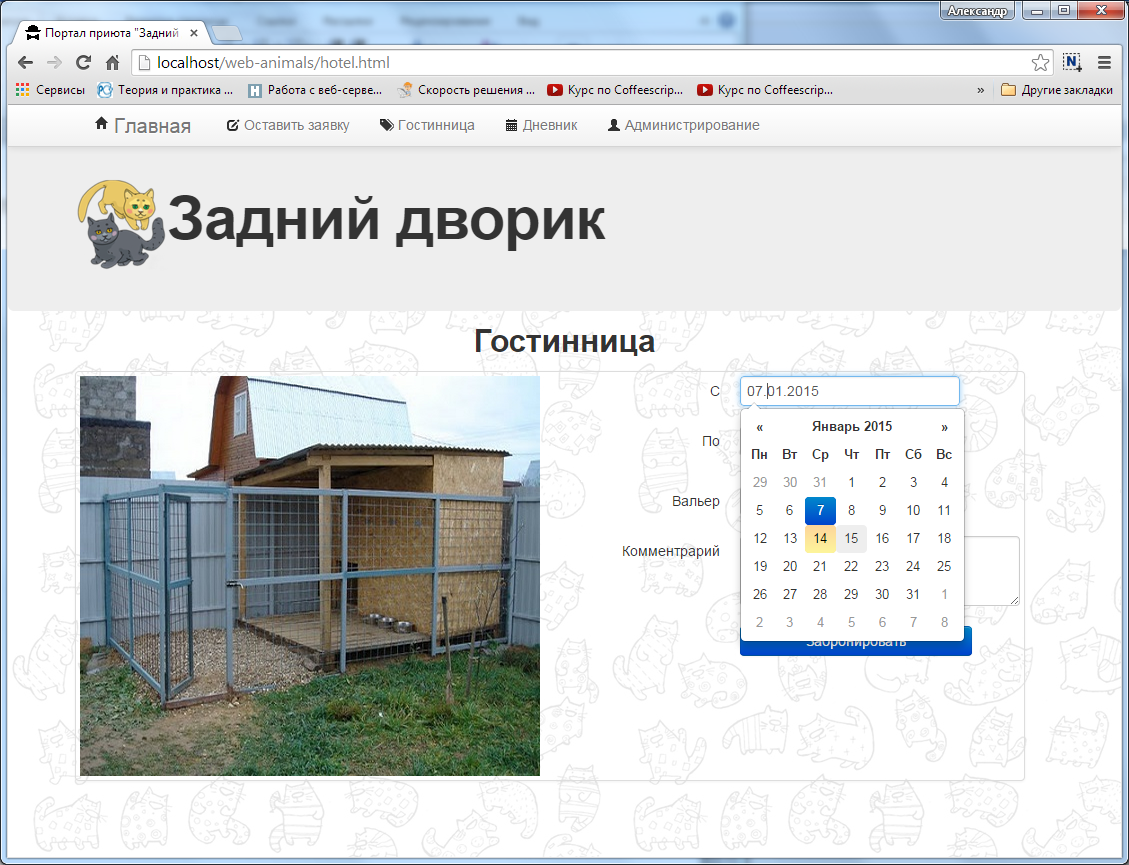
Форма состоит из трех полей ввода – «Фамилия», «Имя», «Контактный телефон». При нажатии на кнопку «Отправить» происходит отправка формы на сервер. Пользователь, при этом получает уведомление о том, что его заявка принята.

**Страница регистрации заявки на размещение животного в гостевом вольере приюта**

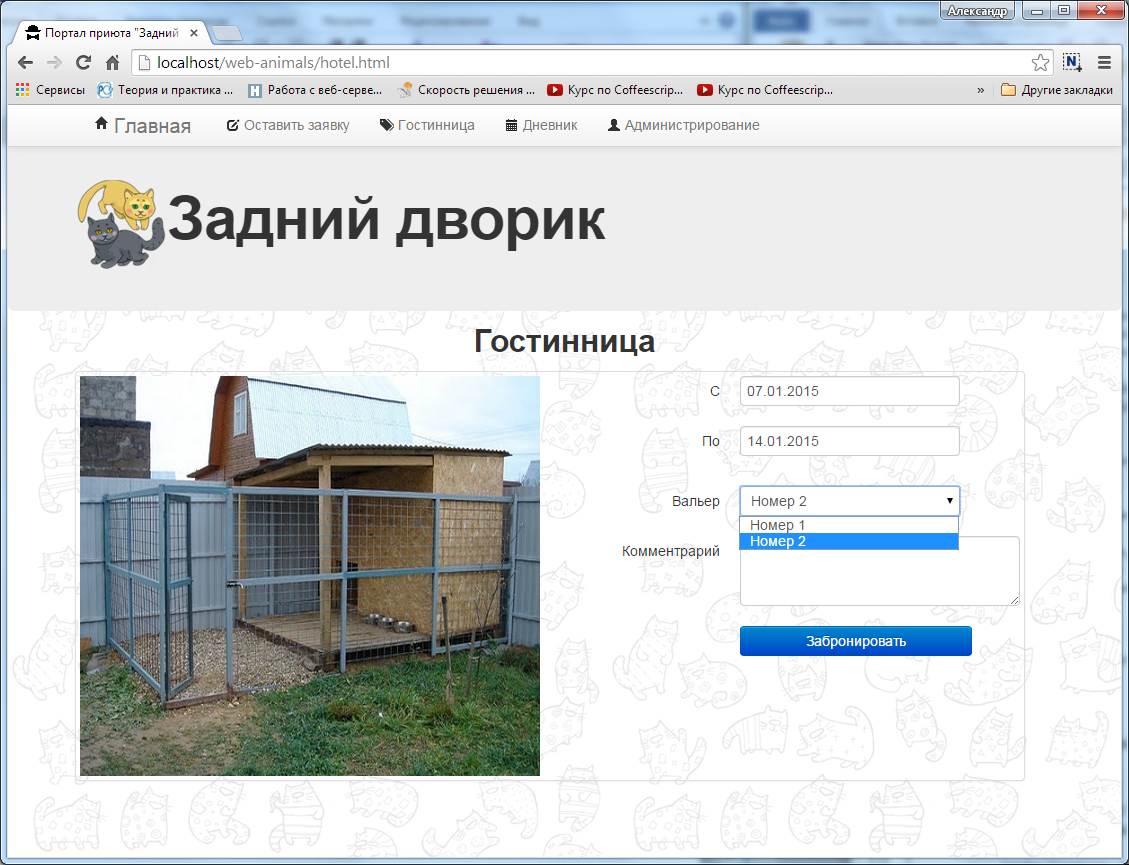
Для перехода на страницу регистрации заявки размещения питомца в гостиничный вольер пользователю необходимо выбрать пункт главного меню «Гостиница». При этом в блок контента будет загружена форма регистрации заявки.



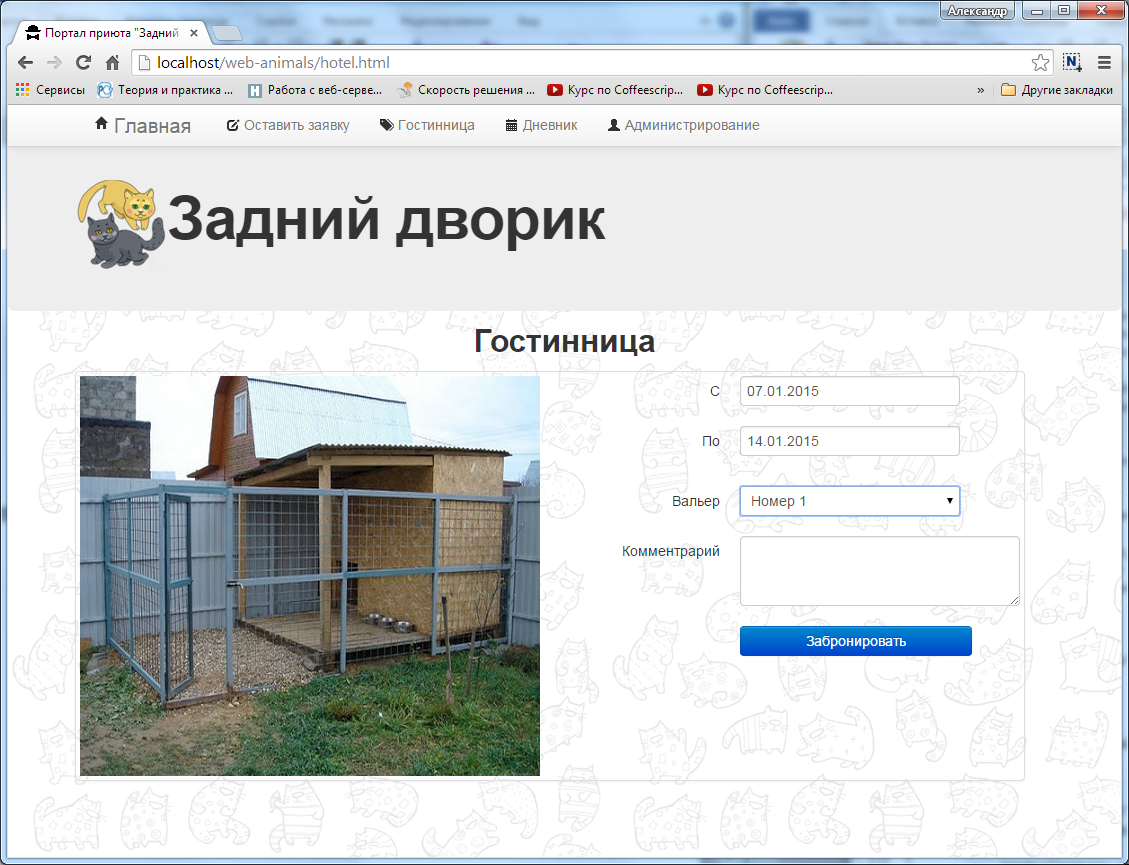
Форма включает в себя два поля для указания интервала дат размещения питомца. Для указания даты в этих полях доступен выпадающий календарь.

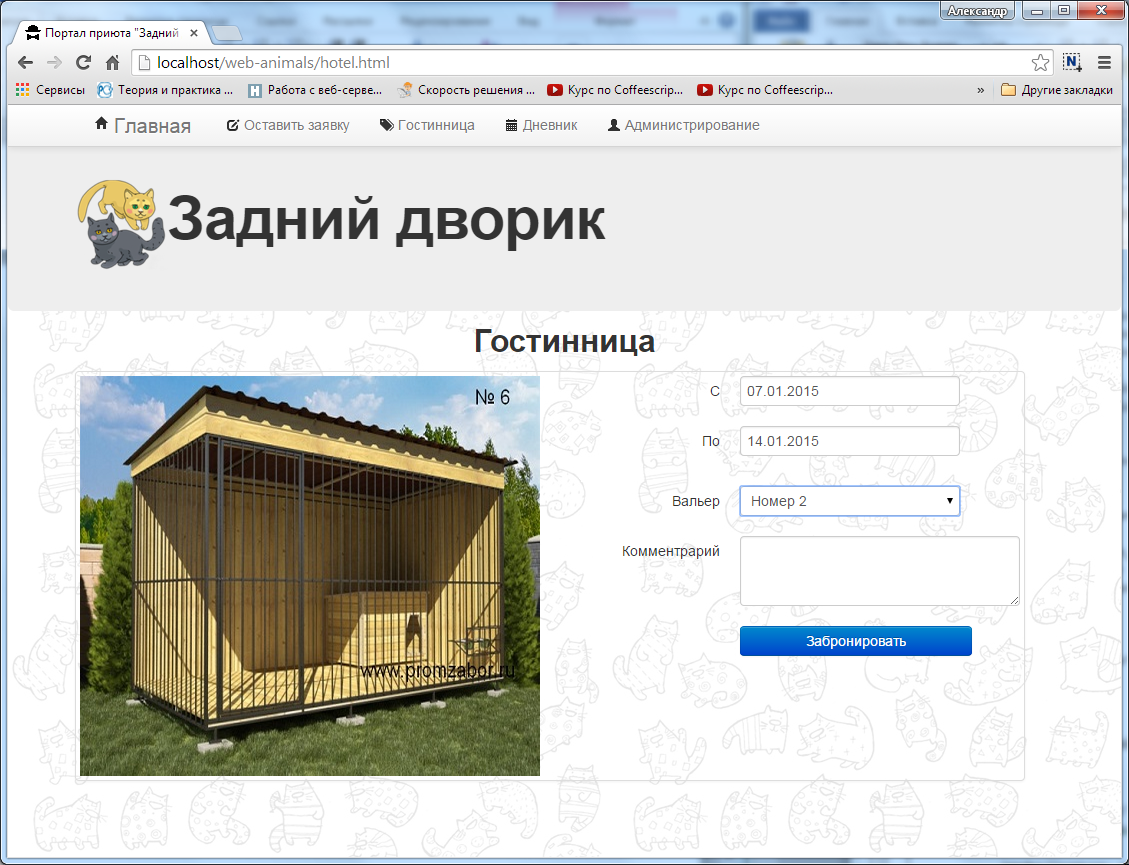


После выбора интервала дат пользователю станет доступен выбор вольера. Список, в поле выбора вольера, состоит только из свободных на указанный период вольеров.

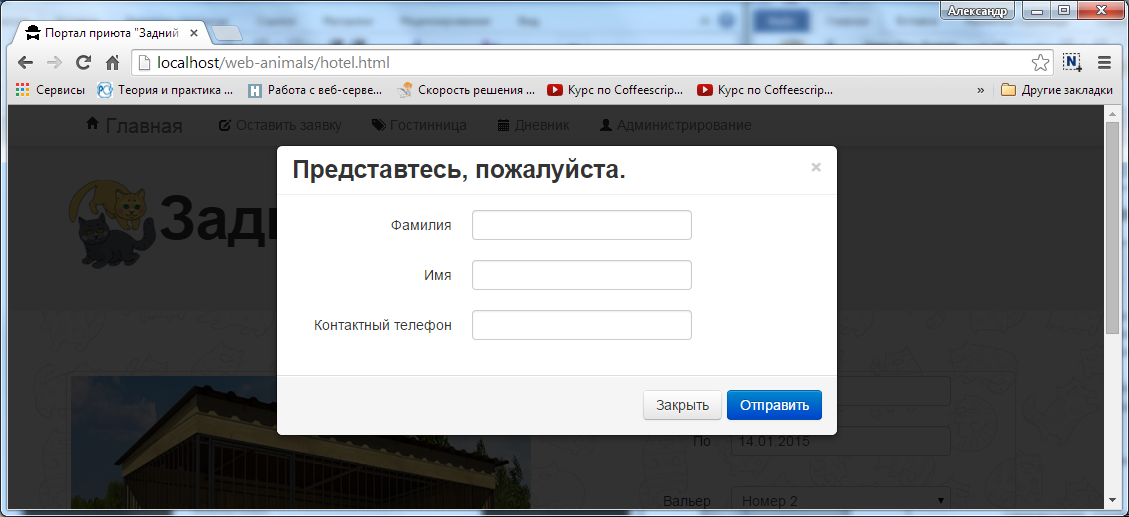


При выборе вольера из списка в блоке слева от поля выбора отображается фотография выбранного вольера.





После нажатия кнопки «Забронировать» пользователю отображается форма ввода контактной информации.



Форма состоит из трех полей ввода – «Фамилия», «Имя», «Контактный телефон». При нажатии на кнопку «Отправить» происходит отправка формы на сервер. Пользователь, при этом получает уведомление о том, что его заявка принята.