*Федеральное государственное бюджетное образовательное учреждение высшего образования*

*«Владимирский государственный университет*

*имени Александра Григорьевича и Николая Григорьевича Столетовых»*

*Кафедра информационных систем и программной инженерии*

***ПОЯСНИТЕЛЬНАЯ ЗАПИСКА***

***к курсовому проекту по дисциплине   
"Технологии программирования"***

***на тему***

*Проектирование и разработка информационно-программной системы «Рекламное агентство»*

*Выполнил: ст. группы ИСТ-119*

*Путинцев С.В.*

*Принял: доц. кафедры ИСПИ*

*Вершинин В.В.*

*Владимир, 2021*

**Аннотация**

Курсовой проект посвящен проектированию и созданию информационно-программной системы «Рекламное агентство». Разработанная информационно-программная система позволяет пользователям получать услуги в сфере рекламы.

Курсовой проект представлен на \_ страницах, рисунков – \_, таблиц – \_, использованных источников – \_, приложений – \_, иллюстрационный материал на \_ листах формата А1.

**ANNOTATION**

The course project is dedicated to the design and creation of an information and software system "Advertising Agency". The developed information and software system allows users to receive services in the field of advertising.

The course project is presented on \_ pages, figures - \_, tables - \_, used sources - \_, applications - \_, illustration material on \_ sheets of A1 format.

**СОДЕРЖАНИЕ**

1. **ВВЕДЕНИЕ**

Данная система предназначена для внедрения или самостоятельного использования на различных сайтах.

Цель работы: разработать информационно-программную систему, которая позволяет пользователям получать информацию об услугах рекламного агентства: стоимость услуг, срок выполнения, примеры работ и т.п.

Для выполнения цели используется платформа .NET, технология ASP.NET, шаблон проектирования MVC, база данных MS SQL.

1. **ПОСТАНОВКА ЗАДАЧИ**

Целью данного курсового проекта является разработка программной

информационной системы «Рекламное агентство».

*Условно, требования к системе разделяются на два вида: функциональные и*

*нефункциональные. Функциональные требования включают в себя список функций, которые должна уметь выполнять система. Нефункциональные требования описывают некоторые особенности будущей системы, к примеру, это может быть указание языка программирования, целевая платформа или платформы, используемые технологии и другие нефункциональные возможности.*

Функциональные требования:

* Авторизоваться в системе
* Посмотреть информацию об агентстве
* Просмотреть справочник менеджеров
* Посмотреть справочник категорий рекламы
* Посмотреть справочник клиента
* Оформление заказа на размещение рекламы
* Снятие рекламы с оборота
* Посмотреть список заказов
* Отменить заказ.
* Оплатить заказ
* Редактировать информацию о заказе
* Удалить заказ из каталога
* Подтвердить заказ
* Посмотреть список пользователей
* Изменить роль пользователя

Нефункциональные требования:

* Реализация на платформе .NET
* Использование технологии ASP.NET
* Использование шаблона проектирования MVC
* База данных MS SQL

1. **ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**
   1. **Описание предметной области**

Рекламное агентство (Ads Agency) представляет собой сервис для

совершения покупок услуг с помощью браузера.

Реклама (Ads) — направление в маркетинговых коммуникациях, в рамках которого производится распространение информации для привлечения внимания

к объекту рекламирования с целью формирования или поддержания интереса к нему. Основными характеристиками являются: тип, цена и срок размещения.

Заказ (Order) объединяет в себе одну и более позиций заказа, а также содержит информацию о клиенте, дате заказа, адресе размещения и текущем состоянии.

Статус заказа (Order status) определяет состояние объекта «заказ», в котором он находится.

* 1. **Выделение и нормализация сущностей**

Для предметной области «Рекламное агентство» были выделены

следующие сущности: клиент, менеджер, администратор, заказ.

Словарь предметной области:

Клиент – человек, пришедший за услугами в агентство.

Атрибуты: ID, Фамилия, Имя, Отчество.

Менеджер – человек, оформляющий клиента.

Атрибуты: ID, Фамилия, Имя, Отчество, ID рабочего места.

Администратор – главный человек, занимающийся ведением документации

и наблюдением за работой сотрудников.

Атрибуты: ID, Фамилия, Имя, Отчество.

1. **СЦЕНАРИЙ ВЗАИМОДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ С СИСТЕМОЙ**

Взаимодействие с системой осуществляется посредствам интернет-браузера. Каждый пользователь имеет одну из нескольких ролей, которые определяют, какими возможностями он обладает. Можно выделить следующие роли и их возможности:

* Клиент
* Посмотреть каталог примеров
* Посмотреть информацию о агентстве
* Посмотреть список заказов
* Отменить заказ
* Оплатить заказ
* Очистить корзину
* Редактировать заказ
* Менеджер
* Посмотреть список заказов
* Найти услугу
* Добавить услугу в каталог
* Редактировать услугу
* Удалить услугу из каталога
* Администратор
* Посмотреть список пользователей
* Изменить роль пользователя
* Просмотр списка оказанных услуг
* Ведение документации

Также, каждый пользователь, может авторизоваться в системе и выходить из своего аккаунта.

Незарегистрированные пользователи могут просматривать каталог, смотреть информацию о услугах и выполнять их поиск, а также зарегистрироваться. При регистрации пользователь автоматически получает роль клиента.

1. **СОЗДАНИЕ ДИАГРАММ UML**
   1. **Диаграмма прецедентов**

На данной диаграмме отображены все прецеденты (бизнес-функции системы) и её актёры (пользователи системы). Каждый прецедент имеет своего инициатора в виде актёра. Некоторые прецеденты входят в отношения включения.

Типичной ошибкой проектирования данной диаграммы является неверная ассоциация актёров и прецедентов.

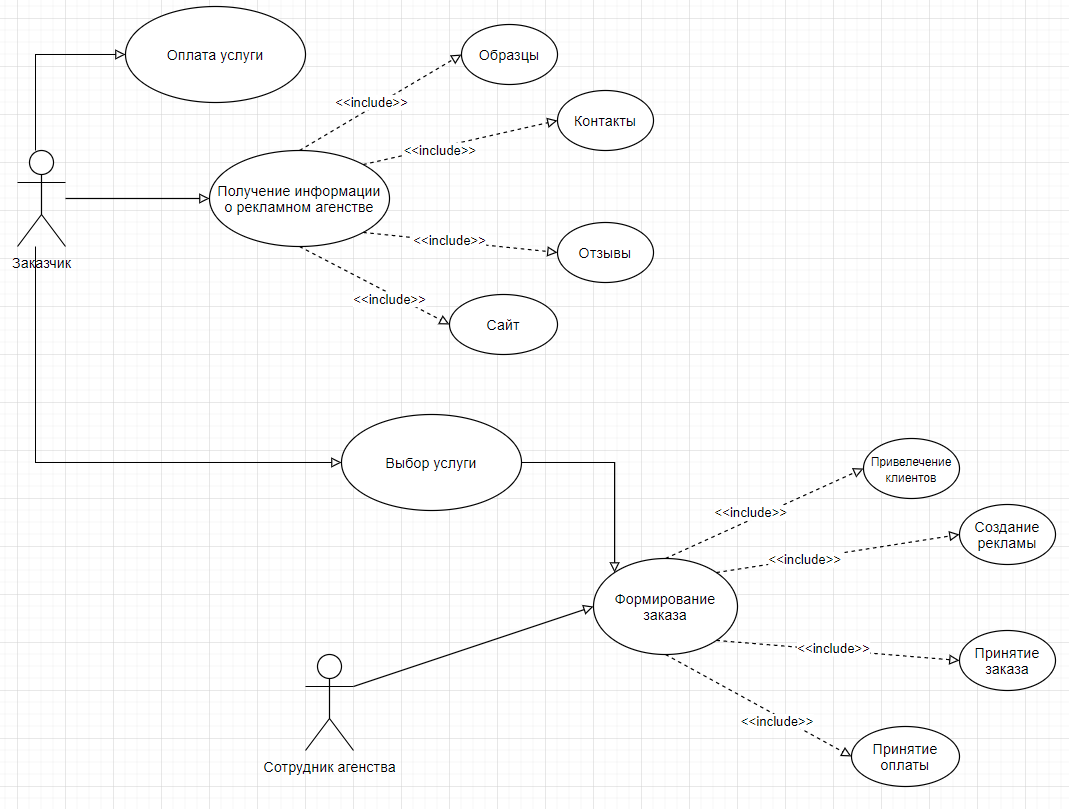


Рисунок 1. Диаграмма прецедентов для клиента и сотрудника агентства

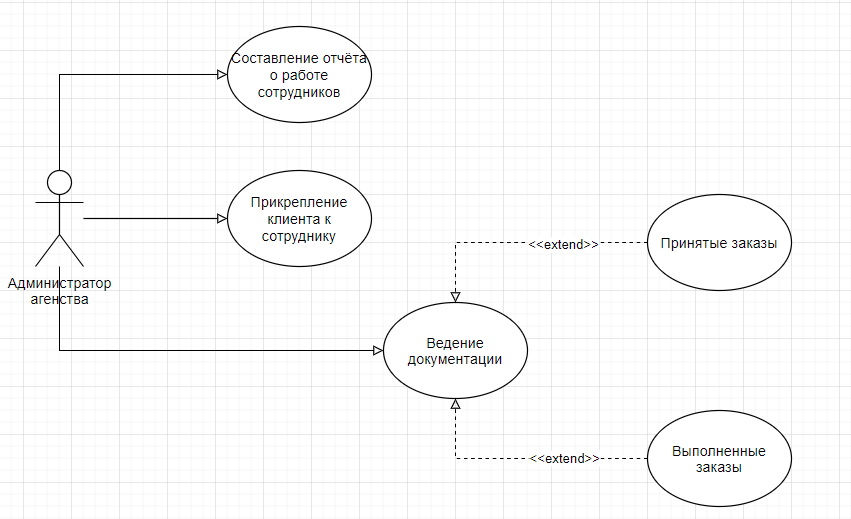


Рисунок 2. Диаграмма прецедентов для администратора

* 1. **Диаграмма состояний**

Диаграмма состояний нужна для изображения состояний, в котором может находится бизнес-объект системы.

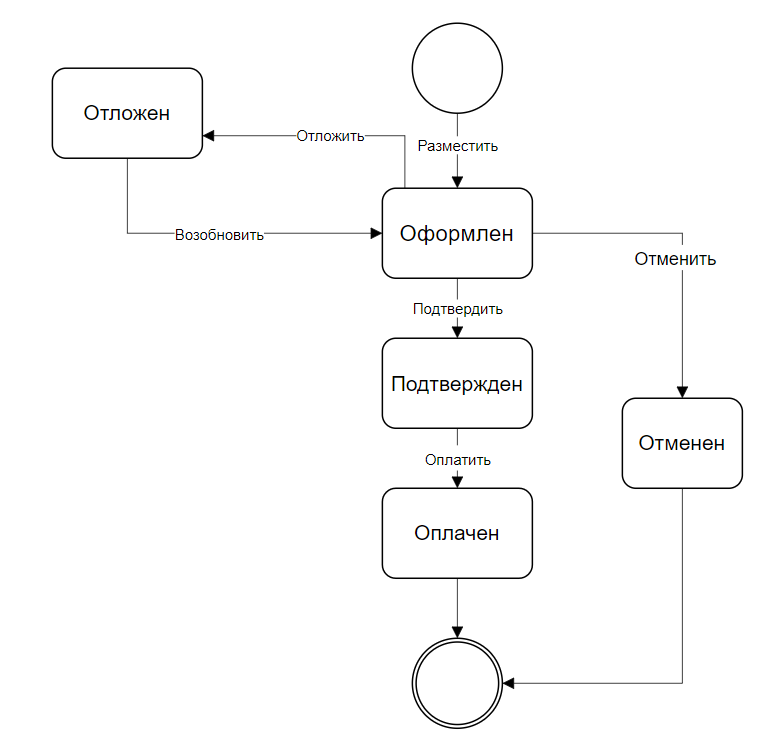


Рисунок 3. Диаграмма состояний заказа

* 1. **Общие принципы организации системы**

Разрабатываемая система представляет собой веб-приложение, построенное на базе архитектуры ASP.NET MVC по шаблону «Интернет-приложение».

Как и в другом приложении, реализующее шаблон MVC – ASP.NET MVC, содержит всё необходимое для реализации всех трёх компонентов: контроллера, модели и представления.

Каждый контроллер относится к своему классу модели и содержит необходимые методы, которые реализуют бизнес-функции и в результате своей работы возвращают пользователю представление, которое отображает результат выполнения операций.

Для реализации модели будет использован Entity Framework.

**6 РЕАЛИЗАЦИЯ МОДЕЛИ**

Entity Framework позволяет использовать любой из трёх способов разработки модели данных: Code-First, Model-First, Database-First.

Code-First. Создание модели начинается с написания кода классов-сущностей, а база данных генерируется фреймворком.

Database-First. Сначала создается база данных, а код сущностей генерируется фреймворком.

Model-First. С помощью встроенного редактора диаграмм создается диаграмма сущностей, которая не привязана ни к коду, ни к базе данных. Код и базу данных генерирует фреймворк.

Для реализации данной системы был выбран подход Database-First. Была создана база данных следующего вида:

\*Сюда поставить диаграмму базы данных\*

После её создания в проекте была создана модель ADO.NET EDM на основе этой базы данных. После того, как код был сгенерирован, его необходимо дополнить нотациями, которые нужны для поддержания валидного состояния модели и отображения её свойств.

Для работы с данными используется экземпляр класса