Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет инженерно–экономический

Кафедра экономической информатики

Дисциплина Средства и технологии анализа и разработки информационных систем

**ПОЯСНИТЕЛЬНАЯ** **ЗАПИСКА**

к курсовому проекту

на тему

**ОНЛАЙН МАГАЗИН**

Студент Пашкевич Б.В.

гр. 572301

Руководитель Хомяков П. В.

Минск 2017

СОДЕРЖАНИЕ

[Введение 5](#_Toc481671244)

[1 Описание предметной области 6](#_Toc481671245)

[1.1 Описание электронных досок объявлений 6](#_Toc481671246)

[1.2 Принципы работы 6](#_Toc481671247)

[2 Описание основных процессов предметной области 9](#_Toc481671248)

[3 Спецификация вариантов использования системы 15](#_Toc481671249)

[4 Информационная модель системы и её описание 17](#_Toc481670352)

[5 Обоснование выбора компонентов и технологий для реализации курсового проекта 20](#_Toc481670353)

[5.1 Spring Framework: Spring MVC 20](#_Toc481671252)

[5.2 Hibernate 21](#_Toc481671253)

[5.3 AngularJS 21](#_Toc481671254)

[5.4 Интернационализация 21](#_Toc481671255)

[5.5 Maven 21](#_Toc481671256)

[6 Модели представления системы и их описание 23](#_Toc481670359)

[6.1 Диаграмма последовательности отправки нового сообщения 23](#_Toc481671258)

[6.2 Диаграмма состояний редактирования информации профиля 23](#_Toc481671259)

[6.3 Диаграмма компонентов 24](#_Toc481671260)

[6.4 Диаграмма развертывания 25](#_Toc481671261)

[6.5 Диаграмма классов 25](#_Toc481671262)

[6.6 Блок схемы алгоритмов, реализующих бизнес-логику системы 25](#_Toc481671263)

[7 Описание применения паттернов проектирования 27](#_Toc481670366)

[8 Руководство по развертыванию системы 29](#_Toc481670367)

[9 Результаты тестирования разработанной системы и оценка выполнения задач 35](#_Toc481670368)

[Заключение 37](#_Toc481671267)

[Список использованных источников 38](#_Toc481671268)

[Приложение А (обязательное) Модели представления электронной доски объявлений 39](#_Toc481671269)

[Приложение Б (обязательное) Блок-схемы методов, реализующих бизнес-логику 44](#_Toc481671270)

[Приложение В (обязательное) Листинг программы 46](#_Toc481671271)

ВВЕДЕНИЕ

В настоящее время интернет становиться все более развитой средой для осуществления коммуникаций с потребителями. В тоже время, существенным является и тот факт, что интернет становиться удобной и достаточно дешевой «торговой площадкой». Все большее количество фирм старается представить свою продукцию в on-line среде. При этом такое представление не ограничивается только лишь созданием промо-сайтов и размещением рекламных баннеров и статей в электронных журналах и на информационных порталах. С развитием интернет-среды развивается и само предложение. Теперь люди могут не только получать интересующую их информацию, но и совершать покупки. При этом с помощью интернет-магазинов можно приобретать товары совершенно разных категорий, как элементарные потребительские, так и высокотехнологичные.

Такое положение вещей обусловлено, во-первых, постоянным, стабильным ростом аудитории пользователей глобальной сети.

Основную часть аудитории сети интернет составляют люди, работающие в офисах и проводящие большую часть времени за компьютером. Как правило, их образ жизни не позволяет им тратить большое количество времени на походы по off-line магазинам в поисках именно того, что им нужно. Возможность заказать интересующий продукт в интернете является для них действительно выходом из ситуации. Интернет существенно ограничивает возможности представления товара, поэтому подходит в большинстве случаев только для повторной покупки, например при приобретении косметики и парфюмерии.

Цель данного курсового проекта – упростить покупку товаров через интернет путем улучшения интерфейса по средствам создания программного продукта. Для достижения поставленной цели необходимо выполнить ряд задач:

* изучить принцип работы электронных магазинов;
* определить список функций необходимых для эффективной работы программного продукта;
* разработать алгоритм работы программы;
* спроектировать базу данных хранимой информации;
* реализовать серверную часть приложения, которая будет реализовывать бизнес логику, и будет выполнять работу с базой данных;
* реализовать клиентскую часть приложения, с удобным и интуитивно понятным интерфейсом;
* протестировать полученное программное средство, и убедиться в его корректной работе.

1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

* 1. Описание онлайн магазинов

Начиная с середины 90-х годов во всем мире наблюдается рост активности в области онлайновой торговли. Вслед за крупными компаниями, производящими компьютерное оборудование в Сеть стали выходить торговцы традиционными товарами. Появилось большое количество книжных магазинов, магазины компакт-дисков и видеокассет, винные магазины. Сейчас практически любые товары можно купить через Сеть.

Эволюция и стремительное развитие IT-технологий, породили такое же быстрое развитие электронной коммерции. Один из популярных on-line бизнесов это интернет-магазин. Даже на фоне мирового финансового кризиса, по различным исследовательским данным, рост составил в среднем в 2009 году от 20 до 35 %. Так же количество покупок, совершаемых через Интернет, ежегодно возрастает в среднем на 20 %. Маркетинговые исследования предсказывают, что к 2020 году количество покупок, которые будут совершаться через Интернет, вырастет более чем на 50 %. Таким образом, интернет-магазин является перспективным направлением бизнеса. Однако данный рынок отличается высокой степенью конкуренции и далеко не все проекты могут выжить. Это происходит из-за того, что зачастую предприниматели, которые начинают своё интернет-дело, не имеют четкого представления об интернет-рынке и о тех трудностях, с которыми они столкнутся в будущем. Как показывают исследования за 2011 год, пока ещё не сформированы общепринятые методы и техника управления интернет-магазином. Поэтому у каждого начинающего свой интернет-бизнес возникал вопрос: как создать онлайн магазин, как управлять интернет-магазином, с чего начинать, чем руководствоваться?

* 1. Принципы работы

Онлайн магазин функционально подобен обыкновенному: это сайт, где каждый платежеспособный желающий может заказать себе товар. Онлайн магазин, как правило, разделен на несколько тематических разделов (категорий), согласно содержанию товаров. Электронные доски объявлений (BBS) предназначены для обмена файлами между пользователями. Они организуются на какой-либо станции (ЭВМ).

Онлайн магазин предоставляет пользователям следующие виды услуг:

* просмотреть список товаров;
* подробно рассмотреть товар;
* выбрать характеристики настраиваемого продукта;
* заказать товар.

Существуют специальные веб-сервисы, работающие с онлайн магазинами. Также существуют и специализированные онлайн магазины, администрация которых придерживается строго тематических товаров по одной или нескольким отраслям промышленности, видам товаров или услуг.

Онлайн магазины можно разделить по видам продаж на:

1. B2B (би-ту-би или «бизнес для бизнеса») — продажа или оказание услуг иным коммерческим предприятиям.
2. B2C (би-ту-си или «бизнес для клиента») — продажа или оказание услуг конечному потребителю.

Существуют следующие виды товаров:

1. Обычный товар – объявления содержат исключительно информацию о приобретаемом товаре, то есть его характеристики, условия покупки и т.д.;
2. Настраиваемый товар – такие товары можно сказать включают в себя информацию о нескольких похожих товарах, а именно их общее описание и различия.

Самое большое количество просмотров приходится на первую страницу сайта, поэтому услуга за размещение рекламы может быть платная, но результативная.

В списках товаров используется их краткое описание для максимального информирования потребителя и экономии места.

А составляя название, необходимо писать его кратко, что бы при запросе поисковым системам было легко найти продукт.

Автоматизация помогает, как и сказано выше, сэкономить время и деньги, так как весь процесс производится на одном месте без каких-либо лишних вложений и действий.

2 ОПИСАНИЕ ОСНОВНЫХ ПРОЦЕССОВ ПРЕДМЕТНОЙ ОБЛАСТИ

Взаимодействие системы с окружающей средой описывается в терминах входа («Данные о товарах»), выхода (основной результат процесса – «Обновленный каталог» и «Заказанный товар»), управления («Правила магазина» и «Нормативные документы») и механизмов («ПО», «Пользователь», «Администратор» – это ресурсы, необходимые для процесса заказа товара).

«Пользователи» – посетители сайта. Они просматривают каталог товаров. Делают заказы.

«Налоговый кодекс» и «Закон о защите прав потребителей» - это правила, которыми управляется процесс функционирования приложения.

В оказании услуг принимает участие «ПО» и «Администратор». Чтобы оформить заказ и получить прибыль.

Основным процессом данного курсового проекта является заказать товар. В соответствии с этим на рисунке 2.1 представлена контекстная диаграмма функциональной модели.

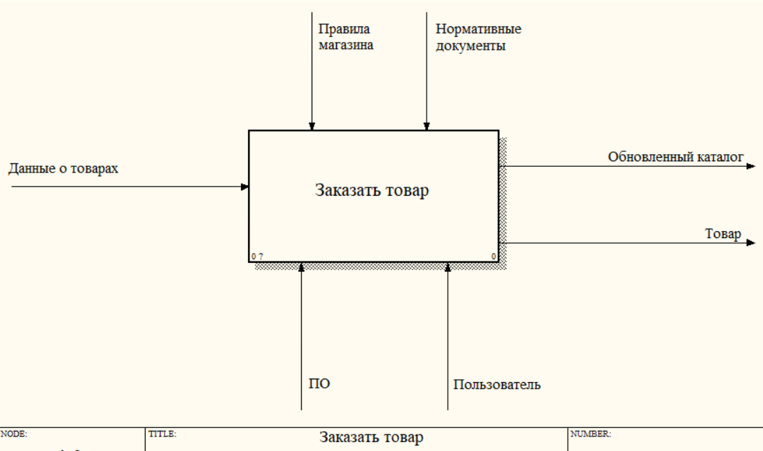


Рисунок 2.1 – Контекстная диаграмма верхнего уровня

В соответствии с рисунком 2.2 представлена диаграмма декомпозиции IDEF0, описывает разделение процесса заказа товара на следующие этапы:

- Завести учетную запись в магазине и авторизоваться;

- Выбрать товар;

- Отправить заявку на заказ товара;

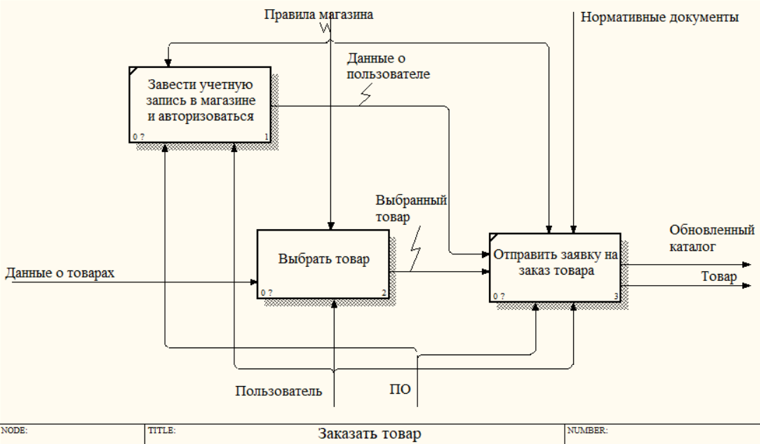


Рисунок 2.2 – Декомпозиция контекстной диаграммы

Весь процесс “Заказать товар” разбивается на:

1)           Завести учетную запись в магазине и авторизоваться – создание учетной записи, для того чтобы можно было заказать товар;

2)           Выбрать товар – представляет собой выбор товара зарегистрированным пользователем, который закрепляется за данным пользователем;

3)           Отправить заявку на заказ товара – оформление заявки;

На рисунке 2.3 представлена декомпозиция процесса «Выбрать товар». Данный процесс состоит из следующих этапов:

- Выбрать категорию;

- Просмотреть товары;

- Выбрать наиболее приемлемый.

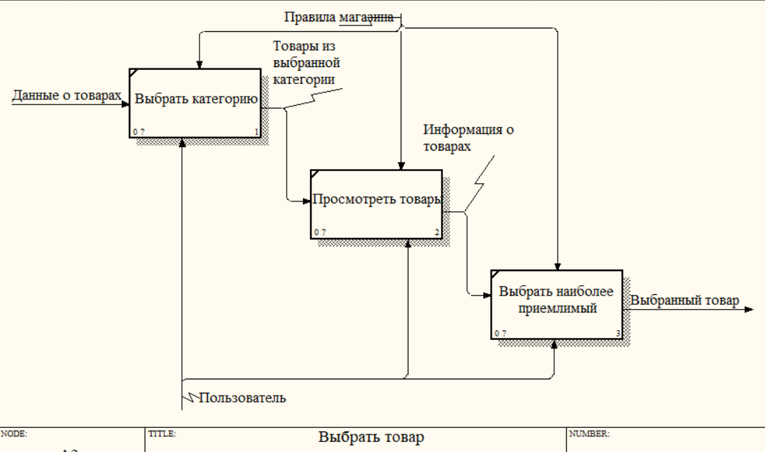


Рисунок 2.3 – Декомпозиция процесса «Выбрать товар»

На рисунке 2.4 представлена декомпозиция процесса «Отправить заявку на заказ товара». Данный процесс состоит из следующих этапов:

- Заказать;

- Обработать заявку;

- Выслать сообщение;

- Обновить каталог.

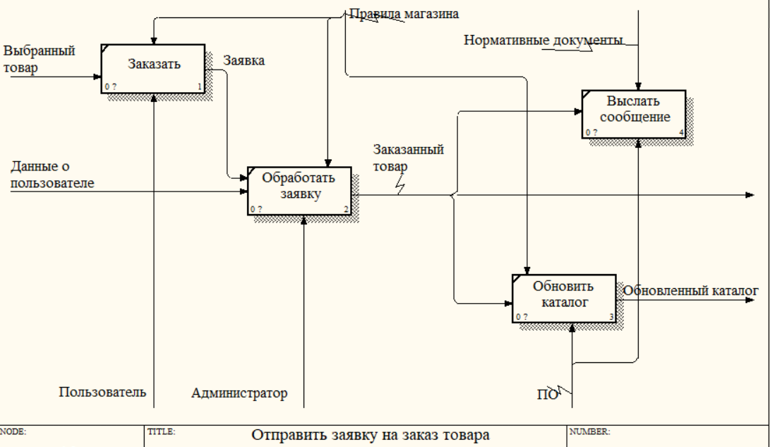


Рисунок 2.4 – Декомпозиция процесса «Отправить заявку на заказ товара»

С помощью данной IDEF-0 диаграммы можно подробно и тщательно рассмотреть процесс заказа товара и его подпроцессы.

3 СПЕЦИФИКАЦИЯ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ

Диаграмма вариантов использования онлайн магазина представлена ниже на рисунке 3.1.

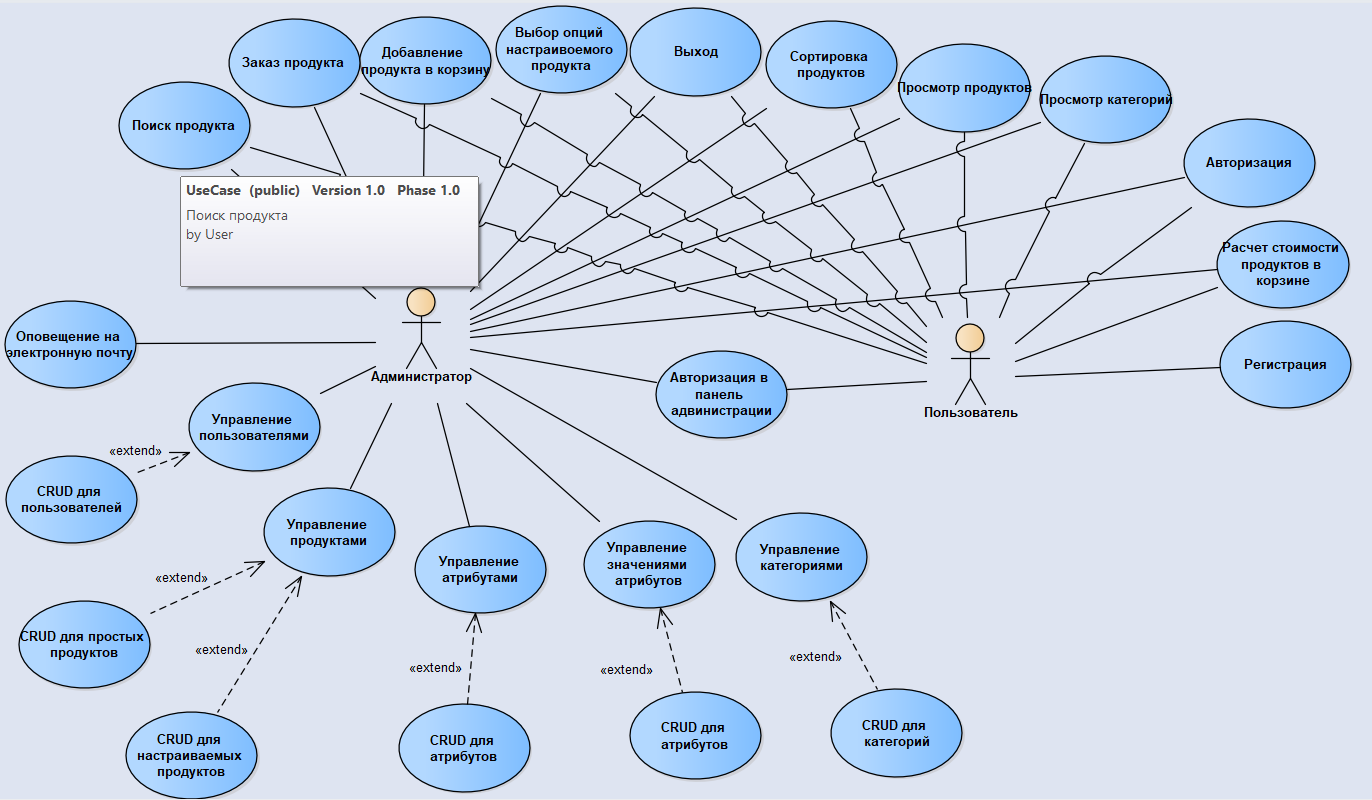


Рисунок 3.1 – Диаграмма вариантов использования

Для организации работы предоставленной системы требуются Администратор и Пользователь.

В основные функции Администратора входит не только контроль за корректной работой системы, но и такие функции как:

1. управление товарами и настройками связанными с ними, что включает их удаление, добавление и редактирование;
2. управление пользователями, а именно изменение имени и роли пользователя, а также их блокирование;
3. управление категориями объявлений, что включает их удаление, добавление и редактирование;
4. оповещение о статусе заказа на электронную почту.

К тому же в системе реализованы общие для администратора и пользователя функции, такие как:

1. авторизация, если аккаунт зарегистрирован;
2. регистрация, если аккаунт в системе отсутствует;
3. Просмотр товаров и категорий;
4. Выбор характеристик настраиваемых товаров.

Касаемо пользователя, в его функции входит не только просмотр товаров, но и поиск, сортировка, добавление продуктов в корзину, заказ товаров и т.п..

4 ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ И ЕЁ ОПИСАНИЕ

Приведение базы данных к нормальной форме – это вопрос проектирования баз данных. И приводить к требуемой нормальной форме базу данных (БД) следует до того, как начнется ее программная реализация, то есть, до того, как начнется само создание БД.

Нормальная форма – свойство отношения в реляционной модели данных, характеризующее его с точки зрения [избыточности](http://zametkinapolyah.ru/zametki-o-mysql/proektirovanie-baz-dannyx-informacionnaya-izbytochnost-izbytochnost-dannyx-v-baze-dannyx-problemy-voznikayushhie-iz-za-informacionnoj-izbytochnosti.html), которая потенциально может привести к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Процесс преобразования базы данных к нормальной форме называется нормализация. Нормализация – это процесс преобразования отношений базы данных к виду, отвечающему нормальным формам. Нормализация предназначена для приведения структуры БД к виду, обеспечивающему минимальную логическую избыточность, и не имеет целью уменьшение или увеличение производительности работы или же уменьшение или увеличение физического объёма базы данных. База данных считается нормализованной, если ее таблицы представлены как минимум в третьей нормальной форме.

Первая нормальная форма(1НФ) говорит, что каждый атрибут отношения должен хранить атомарное значение, каждое отношение должно содержать одинаковое количество атрибутов (столбцов).

Вторая нормальная форма (2НФ) говорит, что отношение находится во второй нормальной форме, если оно находится в 1НФ, и при этом все неключевые атрибуты зависят только от первичного ключа.

Отношение находится в третьей нормальной форме (3НФ), если оно находится во второй нормальной форме и каждый неключевой атрибут зависит только от первичного ключа и не зависят друг от друга.

В результате нормализации были выделены следующие сущности:

* Аккаунт;
* Профиль;
* Объявление;
* Локаль;
* Категория;
* Файл;
* Сущность интернационализации;
* Описание интернационализации.

Логическая модель базы данных показана на рисунке 4.1. Каждая сущность обладает набором атрибутов и обязательным первичным ключом.

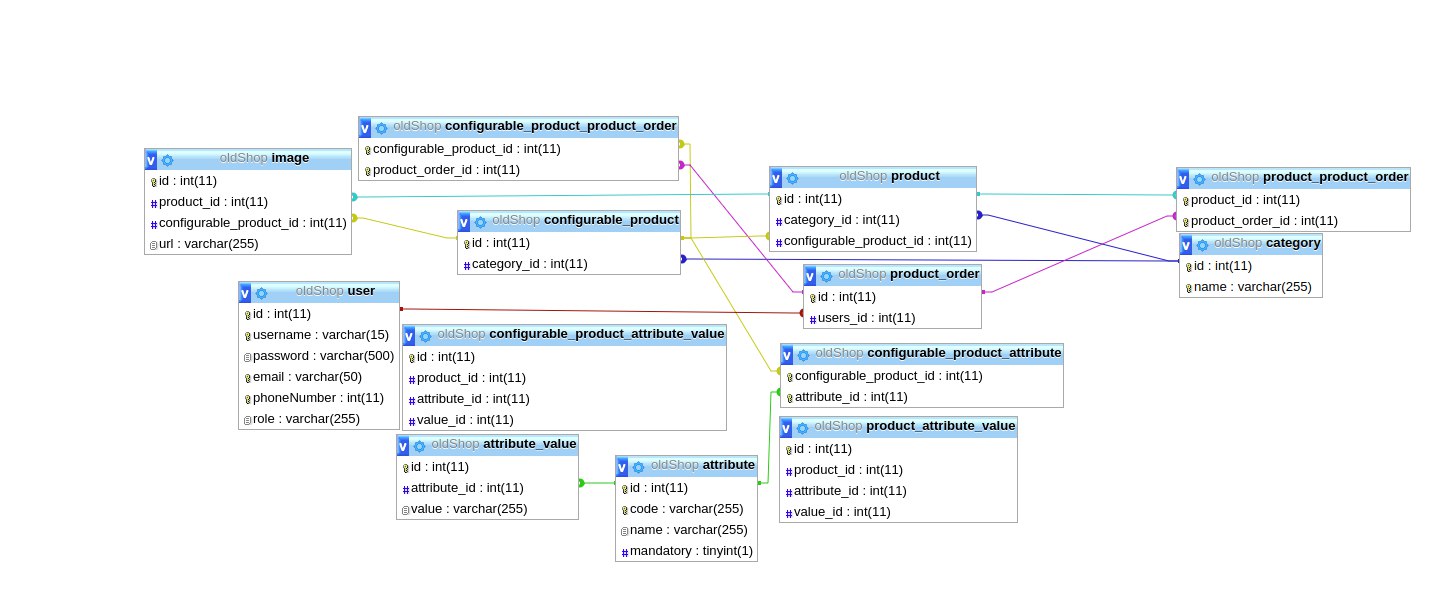


Рисунок 4.1 – Схема базы данных

Рассмотрим сущность Пользователь. Она имеет следующие поля: ИД, роль пользователя, логин, пароль, электронная почта, номер телефона. Поле электронная почта необходимо для того, чтобы администратор мог связаться с пользователем для сообщения статуса заказа. Пароль при сохранении в базу хешируется с помощью стандартного алгоритма UNIX.

Сущность Атрибут содержит имя ИД, код атрибута, название атрибута и поле “mandatory”. Название атрибута – значение которое будет отображаться при просмотре, ИД и код атрибута позволяют его идентифицировать. А поле “mandatory” указывает является ли атрибут обязательным для заполнения.

Далее сущность Значение атрибута, которая характеризуется следующими полями: ИД, ИД атрибута, значение атрибута. ИД атрибута связывает значение с атрибутом которому оно принадлежит.

Очередная сущность - сущность Продукт. Она состоит из поля ИД, ИД категории и ИД настраиваемого продукта. Поле ИД позволяет явно идентифицировать продукт, ИД категории позволяет узнать принадлежность продукта к категории и полу ИД настраиваемого продукта указывает на то принадлежит ли продукт к продуктам созданным по шаблону настраиваемого продукта.

Таблицы находятся в 1НФ, поскольку информация, находящаяся в ячейках неделима.

Так же таблицы соответствуют 2НФ, так как поля, содержащиеся в таблице, зависят от первичного ключа.

Например, в таблице Пользователь поля роль пользователя, логин, пароль, электронная почта, номер телефона никак не связаны между собой, они зависят только от первичного ключа. Такое же состояние у таблицы Атрибут, Продукт и Категория с соответствующими полями, между которыми отсутствует функциональная зависимость.

Из вышеописанного можно сделать вывод, что все таблицы правильно приведены ко 2НФ и 3НФ. Поэтому рассматриваемая БД имеет корректную структуру и проектировку.

5 ОБОСНОВАНИЕ ВЫБОРА КОМПОНЕНТОВ И ТЕХНОЛОГИЙ ДЛЯ РЕАЛИЗАЦИИ КУРСОВОГО ПРОЕКТА

Во время разработки приложения было использовано множество современных технологий, таких как Symfony Framework, Doctrine, jQuery, Composer, Bootstrap, Twig и так далее. Рассмотрим достоинства и необходимость использования некоторых из них по отдельности.

5.1 Symfony

Symfony — свободный фреймворк, написанный на PHP, который использует паттерн Model-View-Controller. Один из самых популярных и современных PHP-фреймворков на данный момент. Он поддерживает множество различных баз данных, в составе фреймворка поставляется Doctrine, Twig, Composer, а также ряд других библиотек. Ключевой особенностью разработки на Symfony является bundle система. Бандл - это совокупность скриптов (php, js и т. д.) и не только (html, css и т. д.)), которые используются для достижения единой цели.

Особенность Symfony в том, что "все есть bundle" - ваше приложение, даже функционал ядра. Обычно бандлы делают независимыми друг от друга, что позволяет упростить разработку проектов путем переноса полностью функционирующих частей из одного проекта в другой.

Symfony использует архитектуру модель-представление-контроллер (model-view-controller), а также разделение объектов, бизнес-логики и внешнего вида приложения. Symfony обеспечивает готовые компоненты, которые могут быть использованы (и используются) для разработки веб-приложений. Все эти компоненты слабо связаны между собой и при желании можно изменить, например, внешний вид приложения, не внося существенные изменения в остальные два компонента.

5.2 Doctrine

Doctrine является инструментом объектно-реляционного отображения для PHP с открытым исходным кодом. Он предоставляет фреймворк для отображения объектно-ориентированной модели данных в традиционные реляционные базы данных. Doctrine не только заботится об отражении сущностей PHP в таблицы БД (и типов данных PHP в типы данных SQL), но также обеспечивает запрос данных и поисковые средства и может значительно сократить время разработки, которое тратится на ручное написание SQL кода.

Doctrine был выбран для работы с БД так как обеспечивает автоматическую генерацию и обновление таблиц и сокращение программного кода, поскольку основные запросы к базе данных представлены как методы. Цель Doctrine – предоставить мощные инструменты, для простого и гибкого взаимодействия с базами данных. Doctrine создаёт слой абстракций над БД и позволяет связать их с концепциями объектно-ориентированного программирования. Работа с базой данных представляет собой исключительно взаимодействие с объектами. Каждая таблица – объект. При необходимости изменения СУБД код переписывать не нужно

5.3 jQuery

jQuery - библиотека JavaScript с открытым исходным кодом для динамических web-приложений. jQuery предоставляет удобный набор инструментов для работы на стороне клиента. Облегчает работу с событиями. Библиотека jQuery содержит функциональность, полезную для максимально широкого круга задач.

5.4 Twig

Twig является шаблонизатором, написанным на языке PHP. Позволяет жёстко разграничить понятия Controller и Viewer, так как при написании шаблона страницы используется только характерные для Viewer функции и синтаксис, который в свою очередь является более удобным и сокращённым по сравнению с PHP.

Twig обладает встроенной системой кеширования страниц, позволяет расширять свой функционал в зависимости от подробностей при разработке проекта путём написания дополнительных классов и методов, является шаблонизатором по умолчанию в Symfony.

5.5 Composer

Composer (getcomposer.org) — это относительно новый и уже достаточно популярный менеджер зависимостей для PHP. Вы можете описать от каких библиотек зависит ваш проект и Composer установит нужные библиотеки за вас! Причём Composer — это не менеджер пакетов в классическом понимании. Да, он оперирует с сущностями, которые мы будем называть «пакетами» или библиотеками, но устанавливаются они внутрь каждого проекта отдельно, а не глобально.

1. МОДЕЛИ ПРЕДСТАВЛЕНИЯ СИСТЕМЫ И ИХ ОПИСАНИЕ

В данном курсовом проекте рассматриваются некоторые модели представления разрабатываемой системы, а именно:

– диаграмма последовательности;

– диаграмма состояния объектов;

– диаграмма компонентов;

– диаграмма развертывания;

– диаграмма классов

Рассмотрим каждую из них детальнее.

6.1 Диаграмма последовательности создания продука

Диаграмма последовательности создания нового продукта отображена на рисунке А.2 Приложения А.

Для начала администратор выбирает пункт «Добавить продукт» на странице продуктов в панели администратора. Далее открывается страница создания продукта, где нужно заполнить данные продукта.

После нажатия на кнопку «Сохранить», информация передается на ProductController.php, откуда данные попадают на сервер. Запрос на создание нового сообщения приходит в ProductAPIController, где обрабатывается в методе addProduct(). Этот метод в свою очередь перенаправляет запрос в

ProductService. И уже сервис обращается к репозиторию, где объявлены методы, которые далее будут обращаться непосредственно к базе данных, в которой хранится все информация. В результате имеем возможность просмотреть созданный продукт.

6.2 Диаграмма состояний авторизации пользователя

Диаграмма состояний изображена на рисунке А.1 Приложения А

Если пользователь хочет авторизоваться, то ему необходимо заполнить форму авторизации(ввести логин и пароль) и нажать кнопку Войти. Далее введенные данные сохраняются в сессию и данные отправляются на сервер для их проверки. В случае, если данные не соответствуют, форма возвращается пользователю с просьбой внести изменения во внесенную информацию, так же в возвращенной форме из сессии получается последнее введенное имя пользователя и заполняется автоматически. Если данные введены верно, то пользователь авторизовывается.

6.3 Диаграмма компонентов

Данная диаграмма представлена в приложении А на рисунке А.3.

На представленной диаграмме компонентов видно разделение работ серверов. На сервере 2 по диаграмме видно, что нету бизнес-логики, сервер 2 предназначен для получения данных с HTML страницы и отправки этих данных на сервер 1. Таким образом сервер2 является связующим звеном между сервером 1 и пользователем. Сервер 1 содержит в себе всю бизнес-логику, так же сервер 1 обращается к базе данных.

6.4 Диаграмма развертывания

Данная диаграмма показывает все присутствующие аппаратные компоненты приложения, а именно сервер 1 на одном персональном компьютере, сервер 2 на другом и база данных на третьем. Данная диаграмма необходима для корректного представления взаимодействия компонентов в распределенной среде. Данную диаграмму можно просмотреть в приложении А на рисунке А.4 .

6.5 Диаграмма классов

В приложении А на рисунках А.5, А.6 и A.7 изображены диаграммы классов для двух модулей системы сервер 1 и сервер 2.

Каждый класс, отображенный на диаграмме, каким-либо образом связан с другим. В связи с тем, что классов очень много, они были разбиты по папкам. В первых двух пакетах находятся классы описывающие бизнес-логику приложения, в пакете Controller классы являются связывающим звеном между бизнес-логикой и пользователем.

* 1. Блок схемы алгоритмов, реализующих бизнес-логику системы

**6.6.1 Схема работы всей программы**

При входе на сайт пользователь попадает на главную страницу, где он может сделать такие действия как “Выбрать категорию товаров”, “Перейти на страницу категорий”, “Выбрать продукт”, “Просмотреть корзину” (Если пользователь авторизован), “Авторизоваться” если пользователь не авторизован и “Выйти” если пользователь авторизован. При просмотре товара если это настраиваемый товар, пользователь может выбрать допустимые характеристики товара. Характеристики выбираются поочередно. При заполнении всех характеристик обновляется изображение товара и становится доступной кнопка “Добавить в корзину”, далее пользователь может “Добавить в корзину” и перейти в нее, где можно “Оплатить товары”, “Очистить корзину” или “Удалить товар”. Далее пользователь может покинуть сайт или вернутся на главную страницу.

**6.6.2 Схема алгоритма добавления продукта в корзину**

Алгоритм, представленный на рисунке Б.2 приложения Б, начинается с того, что пользователь решает добавить продукт в корзину. Сначала он выбирает нужные характеристики, то есть поочередно выбирает из доступного списка значения атрибутов. При заполнении всех характеристик обновляется изображение товара и становится доступной кнопка “Добавить в корзину”, далее пользователь может “Добавить в корзину” и перейти в нее.

7 ОПИСАНИЕ ПРИМЕНЕНИЯ ПАТТЕРНОВ ПРОЕКТИРОВАНИЯ

В ходе проектирования приложения были использованы различные паттерны проектирования.

1. Паттерн MVC (Model View Controller). Блок Модель(Model) инкапсулирует данные приложения. Модуль Представление(View) отвечает за вывод данных пользователю. Контроллер(Controller) отвечает за обработку запросов пользователей и передачу данных модулю View для обработки. В данном проекте моделью является весь пакет service, контроллерами – пакета Controller, а представлениями – пакета views (набор файлов расширения .html.twig).
2. Паттерн Внедрение зависимости (Dependency Injection). Предоставляет возможность внедрения внешней зависимости. Объект перекладывает заботу о создании зависимостей специальному механизму. В проекте файлом кофигурации этого модуля является services.yml, внутри которого указаны все зависимости.
3. Паттерн Сущность-Атрибут-Значение(EAV) – используется для описания сущностей, в которых количество атрибутов(свойств) огромно, но для конкретной сущности применяется лишь несколько из них Для этого в проекте введены такие модели как Attribute и AttributeValue. Так же этот паттерн помогает удобно работать с настраиваемыми продуктами.

Были реализованы паттерны GoF:

1. Паттерн Фасад (Facade). Позволяет скрыть сложность системы путем сведения внешних вызовов к одному объекту. В проекте можно увидеть в классе Methods. Который позволяет обращаться к сторонним сервисам.
2. Паттерн Одиночка – гарантирует создание одного экземпляра класса. Используется в Service Container. Так же предоставляет глобальный доступ к контейнеру.
3. Паттерн Цепочка обязонностей(Chain of responsibility) – преднозначен для организации уровней отвтственности. Используется при роутинге. Изначально запрос идет на главный файл routing.yml, далее направляется в бандл и там уже в нужный файл роутинга.

8 РУКОВОДСТВО ПО РАЗВЕРТЫВАНИЮ СИСТЕМЫ

Все исполняемые команды ориентированы на ОС Ubuntu. При развертывании на другую ОС необходимо следовать пунктам и выполнять команды соответствующие данной ОС.

Устанавливаем Apache2:

sudo apt-get update; sudo apt-get install apache2

Далее устанавливаем PHP версии 5.5.9 и выше. Для этого воспользуемся командой:

sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql

Далее необходимо перезагрузить Apache2:

sudo systemctl restart apache2

Для того, что бы подтянуть зависимости в проекте необходимо установить Composer:

sudo apt-get update

sudo apt-get install curl php5-cli git

curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin --filename=composer

Затем необходимо зайти поочередно в оба проекта и прописать в терминале команду:

composer install

Далее необходимо настроить подключение к БД в файле:

app/config/**parameters.yml.dist**

**Так же необходимо сгенерировать эту базу данных при помощи скрипта.**

После чего необходимо запустить сервера, воспользуемся командой введенной из в терминал из корня проекта:

bin/console server:start

Первым необходимо запустить сервера1(oldShop). После запуска этой команды должно высветится сообщение о том, что сервер запущен, так же symphony автоматически присваивает ему порт 8000. При запуске сервер2(clientOldShop) ему присвоится порт 8001.

9 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ РАЗРАБОТАННОЙ СИСТЕМЫ И ОЦЕНКА ВЫПОЛНЕНИЯ ЗАДАЧ

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта была тщательно и досконально исследована предметная область интернет магазинов, и в результате получено реализованное программное приложение. Поэтому можно сказать, что основные задачи курсового проекта были успешно выполнены.

Приложение предусматривает основные заявленные характеристики. Оно практично и комфортно в эксплуатации, целостно, охарактеризовано и исполнено в рамках заданной предметной области. Приложение хорошо структурировано из-за разбиения на отдельные задачи и подзадачи.

Также были построены различные UML-диаграммы для отображения связей и взаимодействия друг с другом. Поэтому вторичные задачи проектирования проекта тоже были успешно выполнены.

Необходимо заметить, что предметная область детально изучена и проанализирована.

Во время разработки программного продукта были предусмотрены и разобраны возможные ошибки, критические ситуации, которые могут возникнуть при работе с приложением. Чтобы избежать этого, были обеспечены различные проверки и подсказки об устранении ошибок.

В результате, можно утверждать, что заданная цель о сокращении затрат времени пользователем на покупку необходимых товаров успешно достигнута.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Портал предпринимателей [Электронный ресурс]. – Электронные данные. – Режим доступа: http://smallbusiness.ru/work/adv/88

[2] [ZametkiNaPolyah.ru](http://zametkinapolyah.ru/) [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://zametkinapolyah.ru/zametki-o-mysql/normalnye-formy-izbytochnost-dannyx-v-baze-dannyx-tranzitivnaya-zavisimost-proektirovanie-baz-dannyx.html>

[3] Habrahabr [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://habrahabr.ru/post/111102/>

[4] Wikipedia [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.wikipedia.org/wiki/DTO

[5] Крейг Уоллс Spring в действии 4-е издание: Практическое пособие. – ДМК Пресс: Москва, 2013. – 754 с.

[6] Пашкевич, А., Чумаков, О. Современные технологии программирования: Конспект лекций. – Минск: БГУИР, 2007. – 64 с.

ПРИЛОЖЕНИЕ А

**(обязательное)**

**Модели представления электронной доски объявлений**

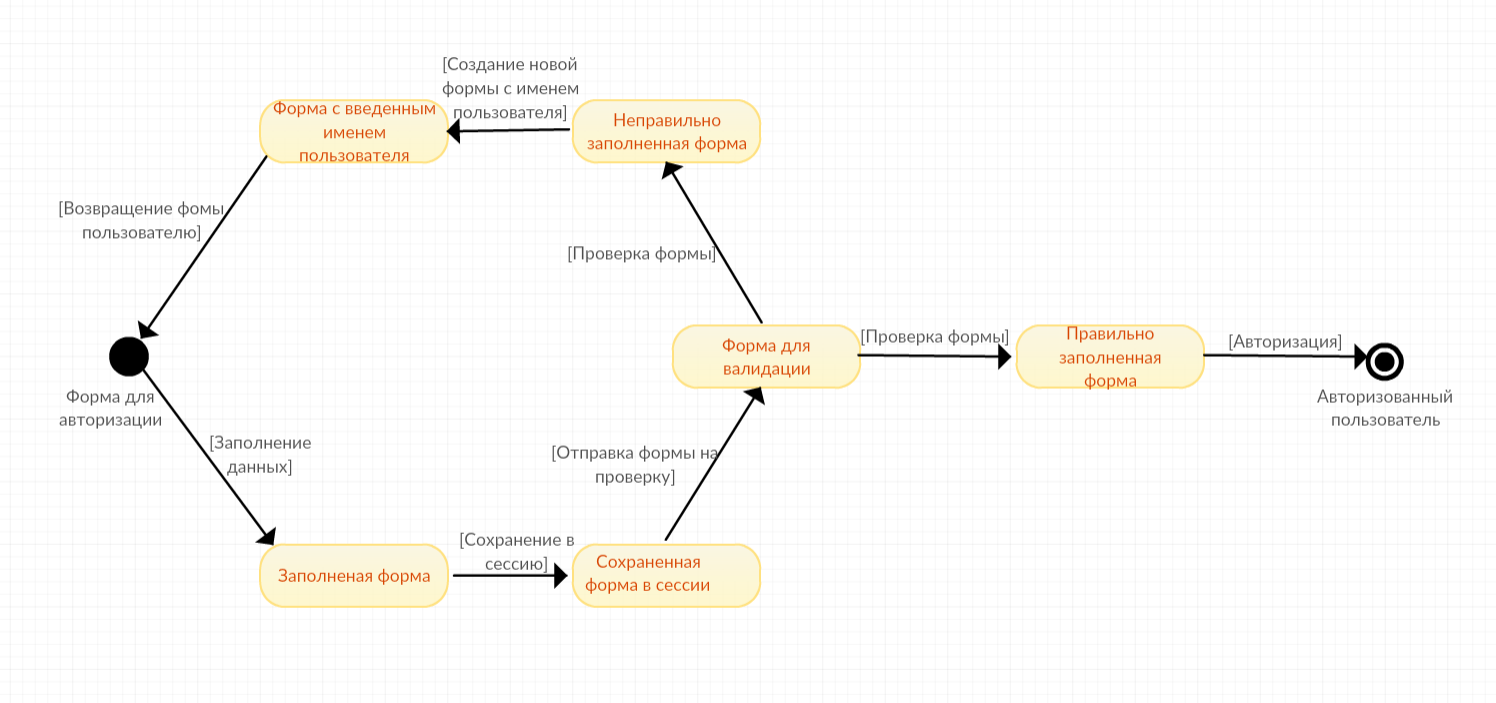


Рисунок А.1 – Диаграмма состояний авторизации пользователя

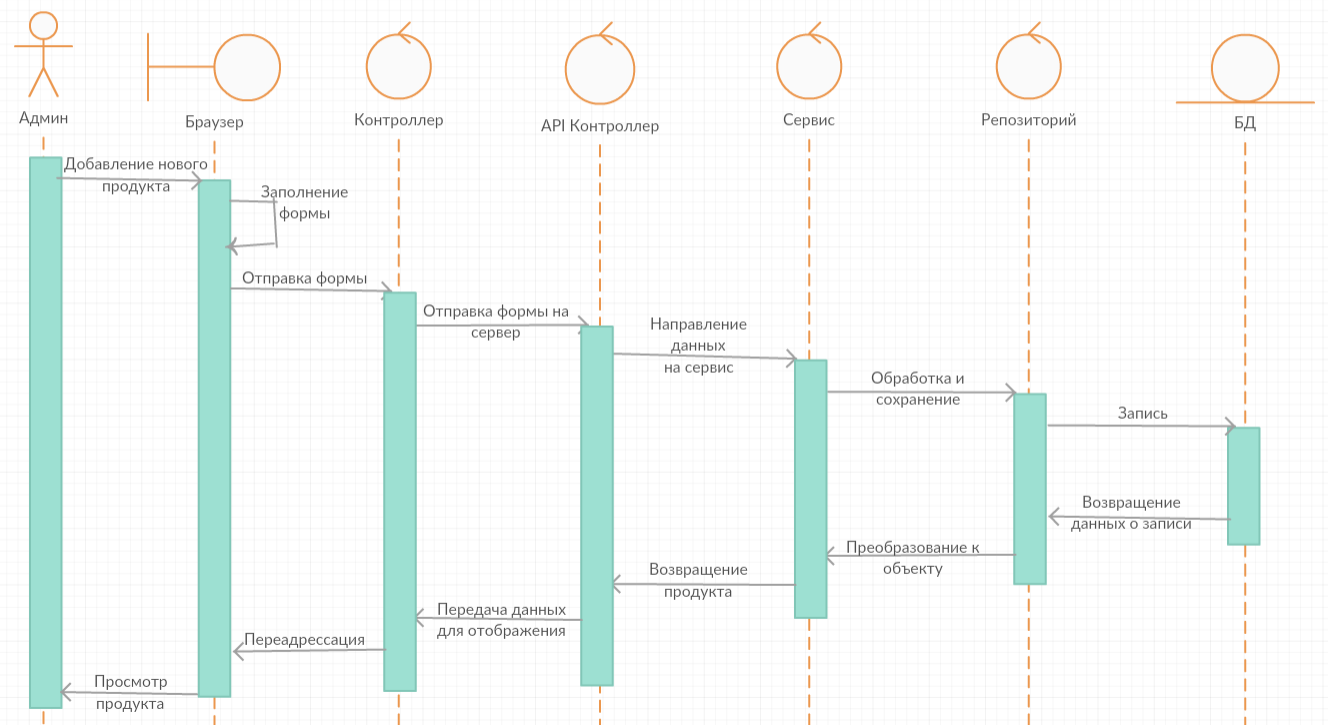


Рисунок А.2 – Диаграмма последовательности добавления нового продукта

Продолжение Приложения А

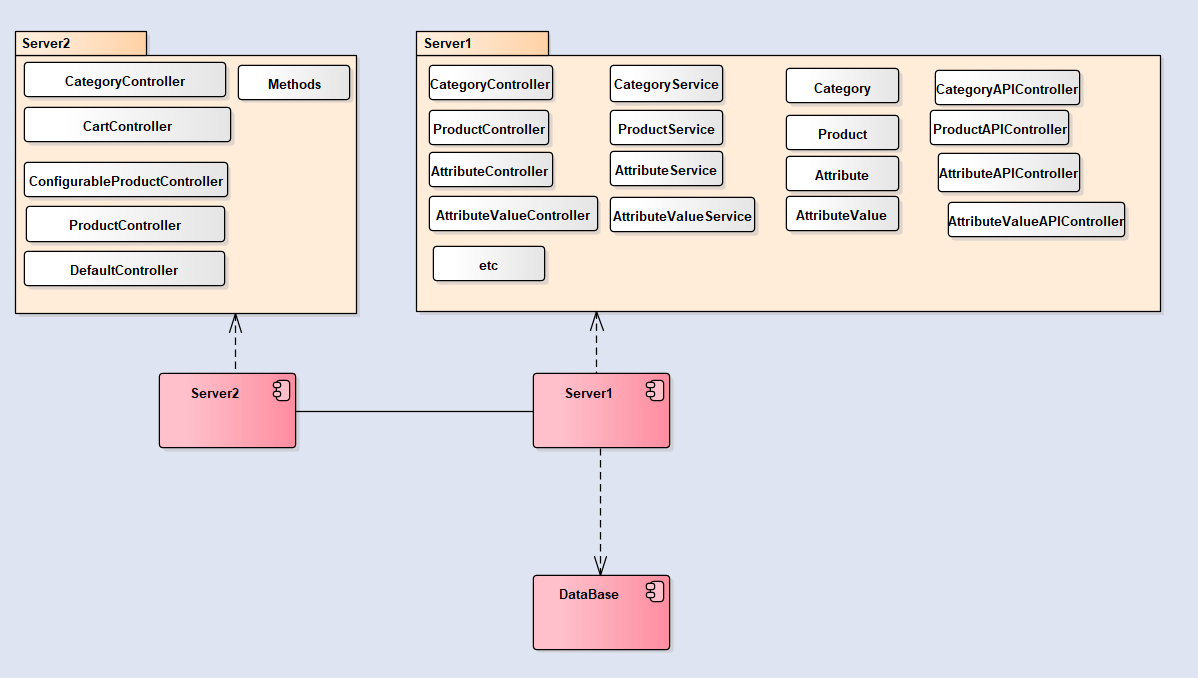


Рисунок А.3 – Диаграмма компонентов модуля базы данных

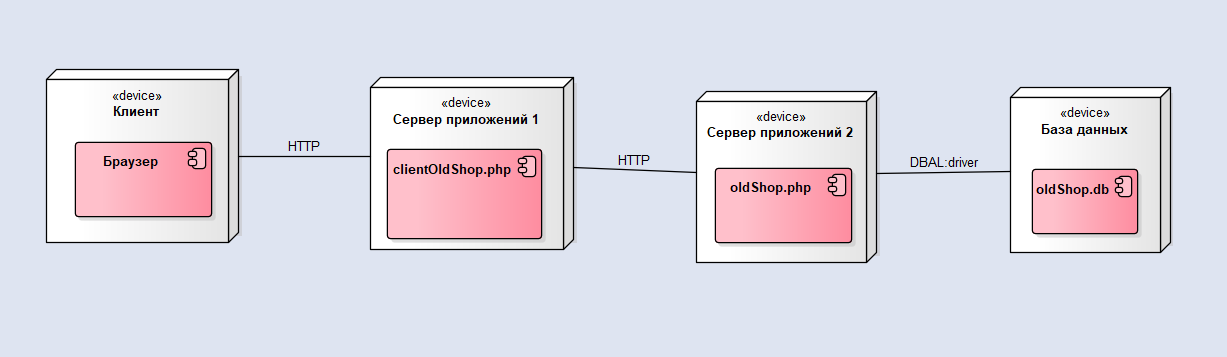


Рисунок А.4 – Диаграмма развертывания

Продолжение Приложения А

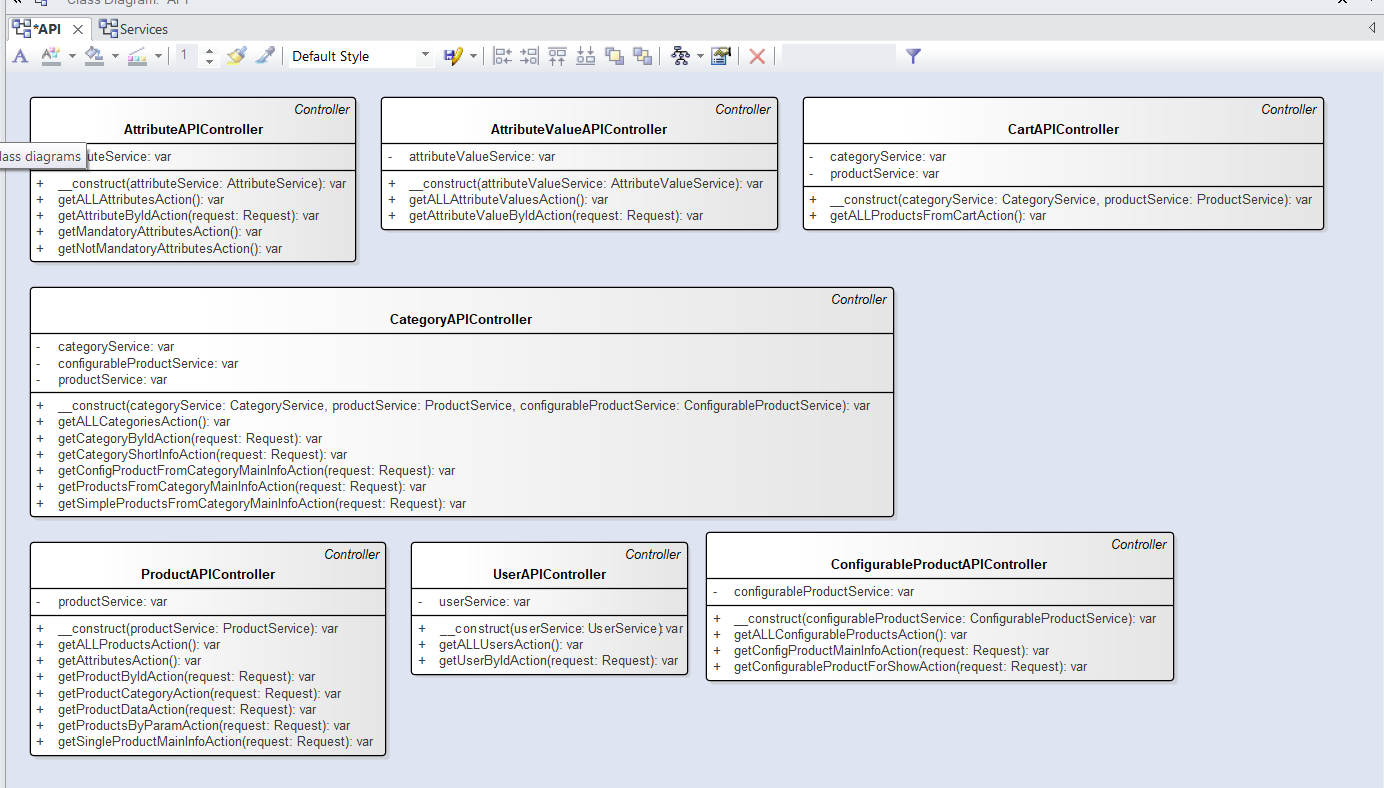


Рисунок А.5 – Диаграмма классов пакета API

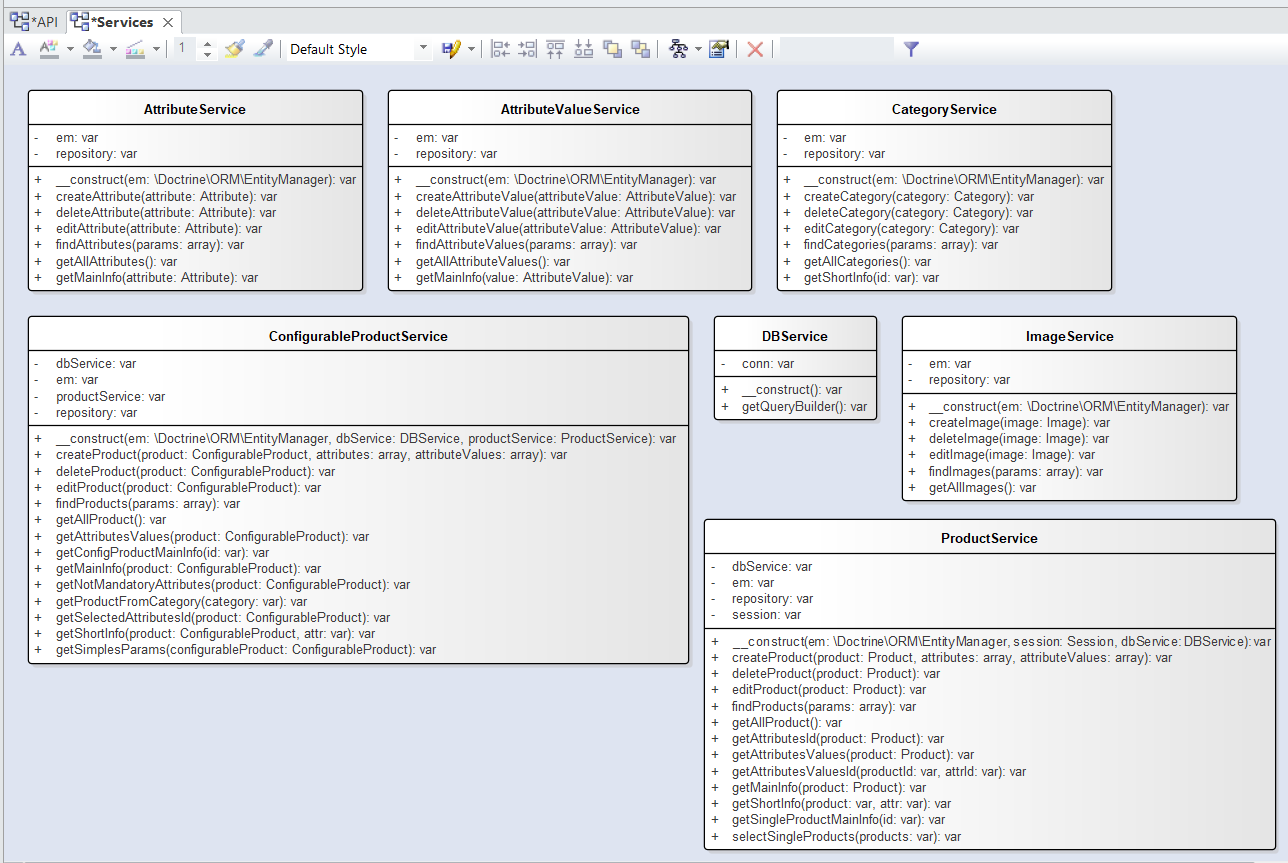


Рисунок А.6 – Диаграмма классов пакета Services

Продолжение Приложения А

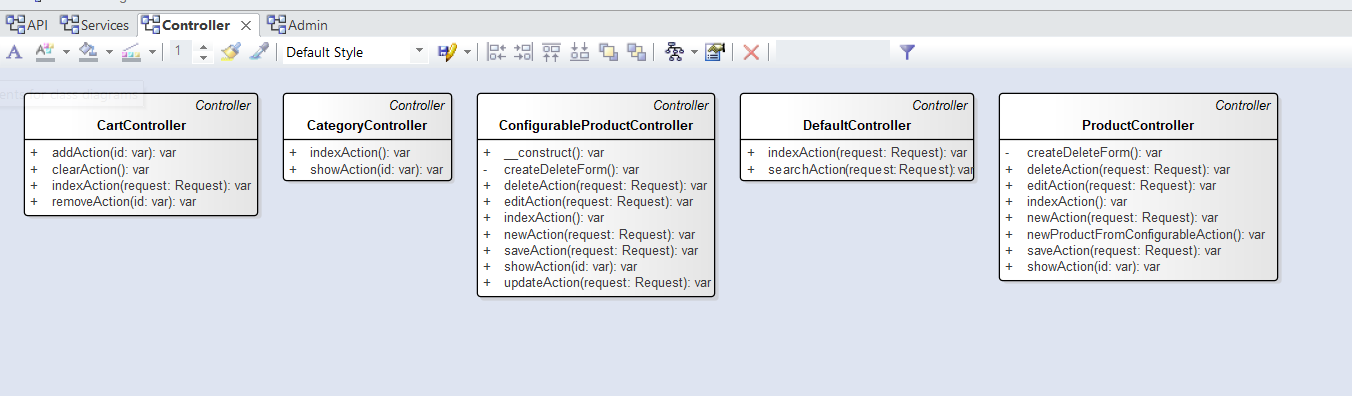


Рисунок А.6 – Диаграмма классов пакета Controller

|  |  |
| --- | --- |
|  |  |

ПРИЛОЖЕНИЕ Б

**(обязательное)**

**Блок-схемы методов, реализующих бизнес-логику**

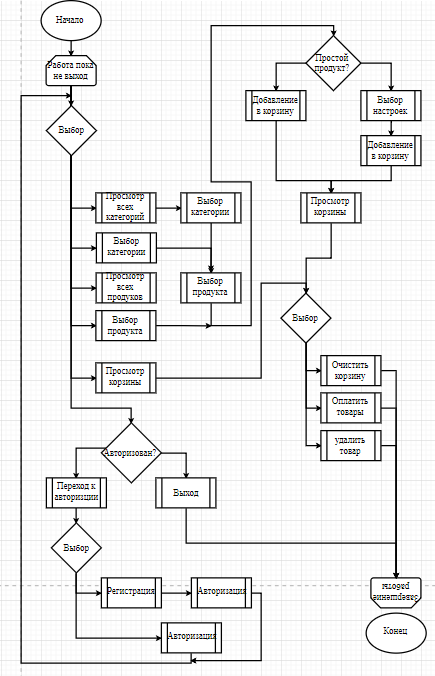


Рисунок Б.1 – Схема работы всей программы

Продолжение Приложения Б

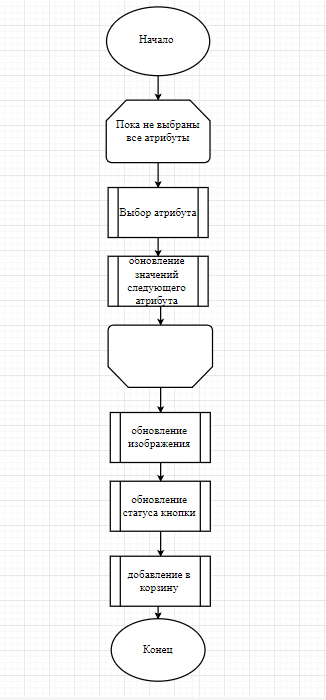


Рисунок Б.2 – Схема добавления в корзину настраиваемого продукта

ПРИЛОЖЕНИЕ В

**(обязательное)**

**Листинг программы**

// Скрипт создания базы данных