

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Забайкальский государственный университет»  
(ФГБОУ ВО «ЗабГУ»)  
Энергетический Факультет  
Кафедра информатики вычислительной техники, прикладной математики

## **КУРСОВОЙ ПРОЕКТ**

По: Веб-Программирование

На тему: Сайт «JewelBling»

Выполнил студент группы ИВТ-17-1 Демидов Владимир Юрьевич

Руководитель работы: доцент кафедры, Семигузов Дмитрий Александрович

Чита  
2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Забайкальский государственный университет»  
(ФГБОУ ВО «ЗабГУ»)  
Энергетический Факультет  
Кафедра информатики вычислительной техники, прикладной математики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к курсовому проекту

По: Веб-Программирование

На тему: Сайт «JewelBling»

Выполнил студент группы ИВТ-17-1 Демидов Владимир Юрьевич

Руководитель работы: доцент кафедры, Семигузов Дмитрий Александрович

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Забайкальский государственный университет»  
(ФГБОУ ВО «ЗабГУ»)  
Энергетический Факультет  
Кафедра информатики вычислительной техники, прикладной математики

ЗАДАНИЕ  
на курсовой проект

По дисциплине Веб-Программирование

Студенту Демидову Владимиру Юрьевичу

Специальности: Информатика, вычислительная техника и прикладная математика

1. Тема курсового проекта: Сайт JewelBling
2. Срок подачи студентом законченной работы: 28 декабря 2019 г.
3. Исходные данные к работе: литературные источники, источники в сети интернет
4. Перечень подлежащих разработке в курсовом проекте вопросов:
  - а) разбор теоретической части;
  - б) программирование сайта.
- 5 Перечень графического материала:
  - а) рисунки;
  - б) таблицы.

Дата выдачи задания «13» сентября 2019 г.

Руководитель курсового проекта \_\_\_\_\_ /Семигузов Дмитрий Александрович/

Задание принял к исполнению

«\_\_» \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_ /Демидов Владимир Юрьевич/

## **РЕФЕРАТ**

Пояснительная записка – 47 страниц, 12 – рисунков, 5 – таблиц, 0 – схем, 0 – диаграмм, 0 – приложений.

HTML, PHP, CSS, POSTGRESQL, SQL, ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ, PGModeler, PGAdmin, UX, USER EXPERIENCE, USE-CASE, FRONTEND, BACKEND.

В работе описывается теоретическая часть веб-программирования, а также описание языков веб-разработки, использованных при создании сайта JewelBling. Сайт, разрабатываемый в условиях курсового проекта, был написан на языке разметки HTML, на языке серверного программирования PHP, а также с использованием язык описания внешнего вида CSS. Также к сайту подключена база данных Postgresql, и в некоторых блоках кода PHP присутствуют SQL вставки. Сайт проектировался согласно пяти методам пользовательского опыта (UX).

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b> .....	6
<b>1. Постановка задачи</b> .....	7
<b>2. Предметная область</b> .....	8
<b>3. Функциональная спецификация</b> .....	13
<b>4. Иерархическая архитектура модулей</b> .....	24
<b>5. База данных</b> .....	27
<b>6. Программная реализация</b> .....	36
<b>7. Руководство пользователя</b> .....	41
<b>ЗАКЛЮЧЕНИЕ</b> .....	45
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b> .....	46

## ВВЕДЕНИЕ

В 21 веке, веке стремительного прогресса, просто красиво оформленный текст и картинки на веб-сайтах не являются новшеством. Требования к сайтам на протяжении времени очень изменились, и теперь для успешного представительства компании в Интернет предоставить своим посетителям различные возможности: обратную связь, подписки, рассылки, поиск по сайту и многое другое. При помощи обычного html этого не достичь, ведь html - это всего лишь язык разметки - инструмент для создания гиперссылок, вставки изображений, таблиц и др. С помощью языка html легко и быстро можно сделать форму для отправки какого-либо запроса. Такая форма будет иметь все необходимые атрибуты: поле для ввода текста, кнопку отправки. Однако при нажатии на такую кнопку ничего не произойдет, ведь не был задан сценарий действий, которые следует выполнить, чтобы получить результат. Поэтому создание интерактивных компонентов - это задача уже для веб-программирования.

В данном курсовом проекте рассматривается разработка фронтенда - части, работающей в браузере, с которой непосредственно взаимодействует пользователь. Это динамические интерфейсы, меню, события по действию пользователя обмен данными с серверной частью, в общем, то, что происходит на клиенте. Также рассматривается разработка бэкенда - серверной части сайта, которая не видна пользователям. Сюда относятся: авторизация, хранение и обработка данных, email рассылки и тому подобное.

Курсовой проект создавался с целью получения навыков разработки сайтов, фронтэнда и бэкенда, а также взаимодействия сайта с базой данных.

## **1. Постановка задачи**

Для создания сайта была рассмотрена задача из другого курсового проекта по базам данных. Необходимо отслеживать деятельность ломбарда. Деятельность компании организована следующим образом: В ломбард обращаются различные лица с целью получения денежных средств под залог определенных украшений. У каждого из клиентов необходимо запросить личные данные. Происходит оценка украшения – выставляется цена, комиссионные, дата окончания договора. По истечении даты договора украшение переходит в собственность ломбарда. После перехода украшений в собственность, ломбард может продавать товары по цене меньшей или большей от той, что была заявлена при сдаче (возможность выставления скидок).

На сайте должна быть реализована возможность оставлять фотографию и необходимые личные данные, а также возможность просмотра статуса своей заявки. Дополнительно, необходимо чтобы на сайте можно было просматривать, какие товары можно купить (какие украшения уже находятся в собственности ломбарда).

## 2. Предметная область

В качестве предметной области выбрана вымышленная компания «JewelBling», которая занимается скупкой украшений в залог и продажей украшений в своей собственности. Информационная система данной компании занимается предоставлением услуг по покупке и сдаче в залог украшений.

Процесс сдачи украшения в залог начинается с подачи посетителем сайта заявки на оценку украшения, которое он хочет сдать в залог. После заполнения формы и указания своих личных данных он автоматически становится клиентом ломбарда. Затем заявку рассматривает оценщик, который выставляет цену, которую он готов выдать клиенту на руки, а также свои комиссионные и дату окончания договора. По истечении договора, при условии что клиент привез украшение в ломбард, оно переходит в собственность ломбарда, иначе заявка просто удаляется из базы данных.

Процесс просмотра наличия или отсутствия оценки для заявки начинается с поиска заявок по индивидуальному ключу.

Процесс просмотра товаров ломбарда начинается с первого посещения сайта. С главной страницы сразу можно перейти в раздел категории товаров, и просмотреть все украшения, которые на данный момент находятся в наличии.

Рассматриваемая предметная область разбита на несколько шагов, каждый из которых отвечает за определенную стадию размещения информации в базе данных.

В момент создания заявки все введенные личные данные записываются в базу данных. Если это новый клиент, то необходимо добавить нового клиента в базу данных, и уже для него добавить заявку. Если клиент уже существует – внести в его заявки еще одну.

При поиске оставленных заявок происходит проверка введенного ключа с наличием соответствующей записи среди клиентов в базе данных. При наличии таковой – вывод всех существующих заявок.

Просматривать товары ломбарда может любой посетитель, для этого процесса шаги не предусмотрены.



Автоматизированная информационная система «JewelBling» предназначена для быстрой и качественной обработки, учета и контроля информации, задействованной в данной предметной области.

Под обработкой понимается добавление, изменение и удаление данных о клиентах, украшениях на оценку, украшениях в наличии.

Под учетом подразумевается быстрый поиск информации по оставленным заявкам, присутствующим в базе данных, а также просмотр товаров по категориям. Например, пользователь, оставивший заявку на оценку украшения, должен быстро найти ее по собственному паспорту, а посетитель, впервые посетивший сайт, должен быстро найти нужное ему украшение перейдя в соответствующую категорию.

Контроль должен осуществляться над заявками, оставленными клиентами. Необходимо выставлять цену для украшения, комиссионные и дату окончания договора. Если клиент лично отдал украшение в ломбард, по истечении даты договора он переходит в собственность ломбарда, либо удаляется заявка.

В соответствии с предметной областью система строится с учетом следующих задач:

- Посетители сайта могут анонимно просматривать товары ломбарда, рассортированные по категориям;
- Если посетители хотят узнать сколько будут стоить их украшения, они должны заполнить форму и передать личную информацию, а также загрузить фотографию украшения;
- У посетителей должна быть возможность просматривать свои заявки, и наличие или отсутствие оценки для них (цена, комиссионные, дата окончания договора);
- У администратора должна быть возможность выставлять скидки на определенные товары, а также возможность редактирования, удаления или переноса товаров из категории в категорию.

Базовые сущности данной предметной области, которые образуют структуру проектируемой информационной системы:

На каждого клиента ломбарда заводится своя таблица. Каждый клиент характеризуется следующими параметрами:

**Таблица: Клиент**

- паспорт серия и номер;
- имя;
- фамилия;
- отчество;
- почта.

Ценности, сданные под залог клиентами, могут иметь одинаковые названия, и их может быть несколько. На каждого клиента ломбарда заводится своя карточка со всеми заложенными ценностями, они характеризуются следующими параметрами:

**Таблица: Предмет клиента**

- Индивидуальный номер (id) предмета;
- Категория;
- Название предмета;
- Краткое описание;
- Цена за предмет;
- Комиссионные;
- Срок возврата денег;
- Паспорт владельца;
- Фотография предмета.

После перехода прав собственности на товар ломбард может продавать товары по собственной цене. Из карточки с заложенными ценностями предмет удаляется, и затем переходит в собственный перечень драгоценностей. Цена может меняться несколько раз, и необходимо учитывать все возможные цены для товара.

У ломбарда есть собственный перечень драгоценностей, который характеризуется следующими параметрами:

**Таблица: Предмет в собственности ломбарда**

- Индивидуальный номер (id) предмета;
- Категория;
- Название предмета;
- Краткое описание;
- Фотография предмета;
- Изначальная цена;
- Нынешняя цена.

Для данной информационной системы необходимо предусмотреть следующие ограничения на информацию в системе:

1. Каждый новый запрос на залог предмета, при уже существующих данных о пользователе, создают новое поле данных в личной карточке клиента. Должны совпадать:
  - Паспортные данные;
2. Все поля в форме должны быть заполнены.
3. Фотография необходима для оценки украшения.

С данной информационной системой должны работать следующие группы пользователей:

- Клиент;
- Администратор.

При работе с системой клиент должен иметь возможность решать следующие задачи:

1. Оставлять запрос на залог предмета в ломбарде.
2. Смотреть статус выставления цены на его предмет.

При работе с системой Администратор должен иметь возможность решать следующие задачи:

1. Выставлять цену на предмет, комиссионные, и сроки возвращения предмета при отмене договора.
2. После истечения срока содержания предмета переносить его из личной карты клиента в карту ломбарда, а также назначение новой цены на предмет и скидочных цен, или удаление предмета из личной карты.

Администрация имеет полный доступ к системе – создание и удаление данных о клиентах, редактирование личных карт клиентов, редактирование карты ломбарда. Клиенты могут только оставлять заявки на залог и просматривать статус заявок.

#### **Функциональные возможности:**

- ведение БД (запись, чтение, модификация, удаление в архив);
- обеспечение логической непротиворечивости БД;
- обеспечение защиты данных от несанкционированного или случайного доступа (определение прав доступа).

#### **Готовые запросы:**

- Отображение украшений в зависимости от категории;
- Заявки.

#### **Выбор СУБД и других программных средств.**

Анализ информационных задач показывает, что для реализации требуемых функций для базы данных подходят почти все СУБД (mySQLi, postgresql и др.). Все они поддерживают реляционную модель данных и предоставляют разнообразные возможности для работы с данными. Была выбрана СУБД postgresql.

Анализ информационных задач показывает, что для реализации требуемых функций сайта подходят языки веб-программирования PHP, HTML, а также языки CSS и JavaScript/Java. Были выбраны языки PHP, HTML, CSS и JavaScript.

### **3. Функциональная спецификация**

#### **Цели проекта**

- Предоставление услуг сдачи в залог украшений потенциальным клиентам в режиме онлайн;
- Предоставление списка доступных товаров онлайн.

#### **Задачи проекта**

- предоставить информацию о компании JewelBling и её услугах;
- предоставить каталог объектов компании;
- С помощью дизайна подчеркнуть имиджевую составляющую компании: предоставление картинок товаров, возможность загрузки фотографий при оценке украшений;
- Обеспечить возможность хранения информации в базе данных.

#### **Пользователи сайта**

В перечень основных пользователей сайта входят:

- Текущие клиенты;
- Потенциальные клиенты (целевая аудитория: инвесторы, покупатели, люди, обладающие неким капиталом, мужчины и женщины 18-55 лет);
- Администрация.

#### **Этапы создания сайта**

1. Прототипирование первых главных страниц сайта;
2. Прототипирование уникальных внутренних страниц сайта;
3. Предоставление заказчиком графических и текстовых материалов для сайта (макетов);
4. Подготовка контента для сайта;
5. Разработка визуальной концепции главных страниц;
6. Разработка и утверждение дизайна второстепенных страниц на основе главной;

7. Программирование сайта на выбранных языках веб-разработки (php,html,css,javascript);
8. Тестирование сайта;
9. Публикация в Интернет на домене Заказчика (НЕ ВЫПОЛНЕНА в рамках курсового проекта, так как сайт был создан для обучения, а не для коммерческого пользования).

### **Дизайн сайта**

Дизайн сайта оптимизируется под разрешение экрана монитора от 768 пикселей до 1280 пикселей по ширине, при размерах меньше 768 пикселей (размеры планшетных экранов и смартфонов) дизайн меняется под оптимальный для планшетных устройств.

Дизайн сайта включает в себя: разработку визуальной концепции, прорисовку графического макета главной страницы и макетов внутренних страниц веб-сайта.

### **Программирование сайта**

HTML-вёрстка и программирование всех макетов сайта производится с учетом стандартных требований к клиентскому программному обеспечению. Под стандартными подразумевается:

- OS Windows XP/Vista/7/8/10 или Mac OS Tiger/Leopard/Snow Leopard
- Google Chrome 14 и выше
- Mozilla Firefox 3.6 и выше
- Opera 10 и выше
- Safari 5 и выше
- IE 8.0 и выше

В качестве базы данных была выбрана postgresql.

## **Требования к оборудованию и администратору системы**

Программно-аппаратное обеспечение системы должно обеспечивать возможность обращения до 10000 пользователей в сутки или 500 одновременных обращений.

В качестве WEB-сервера и сервера базы данных рекомендуется использовать один сервер конфигурации: одноядерный процессор Intel Xeon 2,133 MHz, HDD 100 Gb, 1Gb RAM.

Минимальные технические требования к ПО:

- ОС на любой UNIX или Windows-платформе;
- PHP 7.4.1 и выше;
- XAMPP 7.4.1 (Apache) и выше;
- Postgresql 10 и выше.

Для успешной установки и поддержки системы в дальнейшем, администратор сервера должен обладать навыками настройки UNIX-сервера с установленным WEB-сервером.

## **Структура сайта**

Структура сайта дублируется основным меню навигации, поиском заявок и нижним колонтитулом, и состоит из следующих пунктов:

1. Домашняя страница
  - a. Краткое описание компании;
  - b. Три самые популярные категории товаров;
2. О сайте
  - a. Более подробное описание деятельности сайта и компании;
  - b. Ссылка на заполнение формы для сдачи украшений в залог;
3. Категории товаров
  - a. Конкретные категории
    - i. Полный список товаров конкретной категории;
  - b. Ссылка на заполнение формы для сдачи украшений в залог;

4. Сдать в залог
  - а. Бланк с личной информацией сдающего в залог, фотографией украшений и другой информацией;
5. Скрытая страница авторизации для администрации.

### **Типовые элементы сайта**

Каждая страница сайта состоит из типовых элементов, повторяющихся из раздела в раздел. К типовым элементам можно отнести:

1. Шапка и основное меню навигации;
2. Поиск заявок;
3. Ссылка на подачу заявки;
4. Нижний Колонтитул.

### **Шапка и основное меню навигации**

Шапкой сайта называется типовой информационный блок, обычно размещаемый в верхней части каждой страницы.

1. Логотип сайта;
2. Основное меню навигации
  - а. Домашняя страница
  - б. О сайте
  - с. Категории товаров
  - д. Сдача в залог

### **Поиск заявок**

Расположен после шапки с основным меню навигации. Это блок, в который клиент может ввести свои данные для проверки статуса своих заявок.

### **Ссылка на подачу заявки**

Расположена сбоку справа, или для маленьких экранов (меньше 768 пикселей) – снизу. При нажатии на кнопку переносит посетителя сайта на страницу для заполнения бланка и сдачи товара на оценку специалисту.



## Нижний Колонтитул

Находится в самом низу. Содержит информацию об авторских правах, лицензии, пользовательские соглашения и важную юридическую информацию.

## Цветовая схема сайта JewelBling








Цвет	Нех код	Название переменной в root	Описание	В каких частях сайта используется
	<b>#f4f4f4</b>	--backColorPrimary	Светло-серый основной цвет заднего плана.	body
	<b>#354248</b>	--backColorSecondary	Темно-серый второстепенный цвет.	.dark
	<b>#cccccc</b>	--backColorGray	Чистый серый цвет, акцент, наведение.	#boxes .box ul#services li
	<b>#e6e6e6</b>	--backColorBrightGrayPale	Главный цвет квадратилов.	#boxes .box ul#services li
	<b>#d8d8d8</b>	--boxColorBrightGrey	Цвет выделения, акцента квадратилов.	#boxes .box: hover #main-col .box:hover
	<b>#e8491d</b>	--coralPrimary	Второстепенный дополняющий цвет.	.button1 .button2 header header .highlight header .current a services h2
	<b>#f25044</b>	--coralPaleSecondary	Цвет наведения, акцента.	.button:hover .button2:hover

Таблица 3.1 – Цветовая схема сайта JewelBling.

## Типографика сайта JewelBling

Пример	Название	Описание	В какой части сайта используется
<p>Aa Ee Rr 0123456789</p> <p>Aa Ee Rr 0123456789</p> <p>Aa Ee Rr 0123456789</p>	'Raleway'	Компьютерный свободный шрифт с поддержкой кириллицы, относящийся к типу шрифты без засечек.	body
<p>Aa Ee Rr 0123456789</p> <p>Aa Ee Rr 0123456789</p> <p>Aa Ee Rr 0123456789</p>	Arial (Ариал)	Компьютерный шрифт класса неогротеск, семейства Helvetica, относящийся к типу шрифты без засечек.	body
<p>Aa Ee Rr 0123456789</p> <p>Aa Ee Rr 0123456789</p> <p>Aa Ee Rr 0123456789</p>	Helvetica (Гельветика)	Компьютерный шрифт класса неогротеск, широко используется в логотипах различных компаний, относящийся к типу шрифты без засечек.	body
<p>Aa Ee Rr 0123456789</p> <p>Aa Ee Rr 0123456789</p> <p>Aa Ee Rr 0123456789</p>	Sans-Serif (рубленый шрифт, гротеск)	Чаще всего используется для привлечения внимания в заголовках, относится к типу шрифты без засечек.	body

Таблица 3.2 – Типографика сайта JewelBling.

## Use Case Диаграмма

Диаграмма вариантов использования (use case diagram) отражает отношения между актерами и прецедентами и является составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

При моделировании системы с помощью диаграммы прецедентов необходимо:

- чётко отделить систему от её окружения;
- определить действующих лиц (актёров), их взаимодействие с системой и ожидаемую функциональность системы.

Для отражения модели прецедентов на диаграмме используются:

- рамки системы (англ. system boundary) — прямоугольник с названием в верхней части и эллипсами (прецедентами) внутри. Часто может быть опущен без потери полезной информации,
- актёр (англ. actor) — стилизованный человечек, обозначающий набор ролей пользователя (понимается в широком смысле: человек, внешняя сущность, класс, другая система), взаимодействующего с некоторой сущностью (системой, подсистемой, классом).
- прецедент — эллипс с надписью, обозначающий выполняемые системой действия (могут включать возможные варианты), приводящие к

наблюдаемым актёрами результатам. Надпись может быть именем или описанием (с точки зрения актёров) того, «что» делает система (а не «как»).

**Тема:** сайт ломбарда.

**Задача сайта:** демонстрация товаров пользователям онлайн, возможность оставлять заявки на оценку украшений онлайн, просмотр статуса заявок.

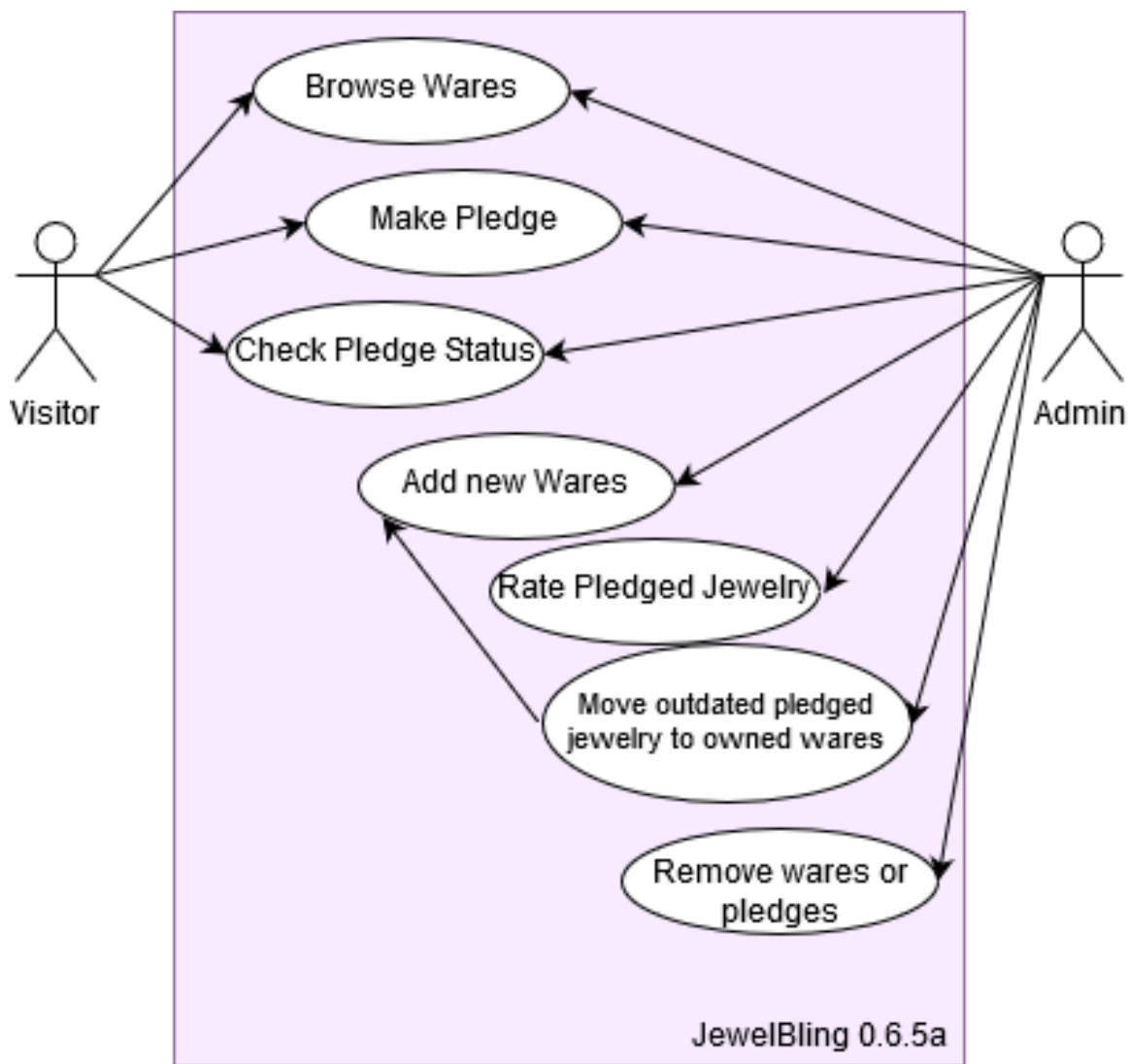


Рисунок 3.1 – Use-case диаграмма. Слева – посетитель, справа – администратор. Посетитель может просматривать товары, оставлять заявку, проверять статус заявки. Администратор может то же что и посетитель, но еще и может добавлять новые товары, оценивать украшения в заявках, переносить украшения из категории в категорию, удалять украшения.

## **Макет сайта**

Есть большая разница между визуальным образом, разработанным с учетом технических возможностей HTML и CSS, и вайерфреймом, представляющим собой простой эскиз. Макет сайта – это только весь дизайн. Он определяет порядок всех элементов ресурса, цветовые оттенки, формы, конкретное расположение компонентов по отношению друг к другу. Создание макета необходимо для создания качественного сайта, и является промежуточным этапом на пути к конечному продукту.

На начальной стадии создается каркас веб-сайта, известный как схема страницы или экрана. Он включает в себя элементы, которые планируется разместить на сайте, и их расположение. Затем определяется стиль, цветовое решение, форма и дизайн всех элементов. Разрабатывая макет нужно представить насколько эргономичным будет проект, а также внешний вид страницы в целом.

Прототип – конечный интерактивный продукт. Это готовая работа, обладающая набором функций, например переходом между страницами. Прототип нужен в качестве обоснования концепции.

Макет и прототип полностью рабочие, с ссылками-переходами на другую главную страницу. Реализовано на сайте [figma.com](https://figma.com)

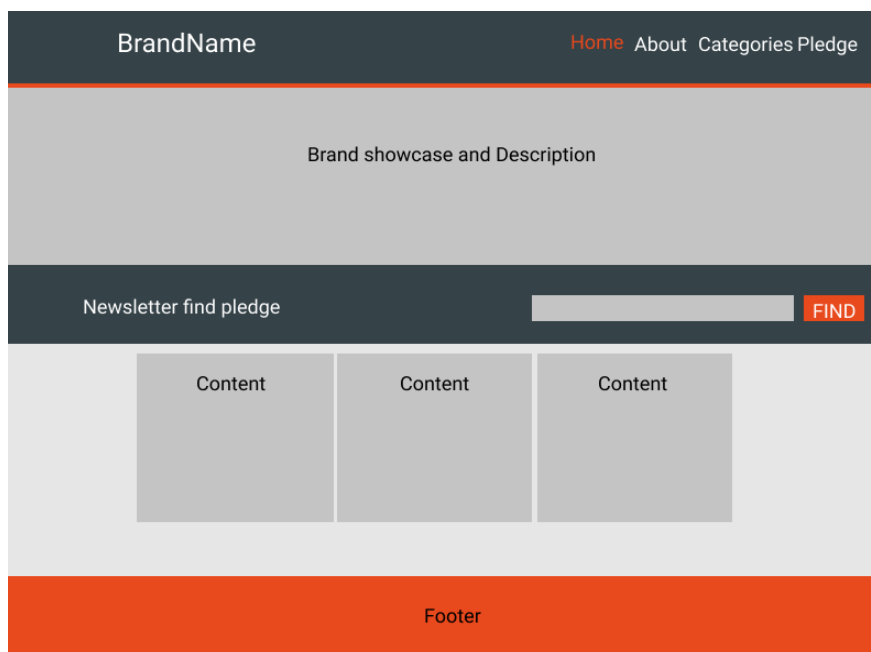


Рисунок 3.2 – Макет сайта, домашняя страница. BrandName – название бренда, Brand showcase and Description – фотография бренда и описание, Newsletter find pledge – найти заявку на залог, Content – популярные категории, Footer – закрывающая часть.

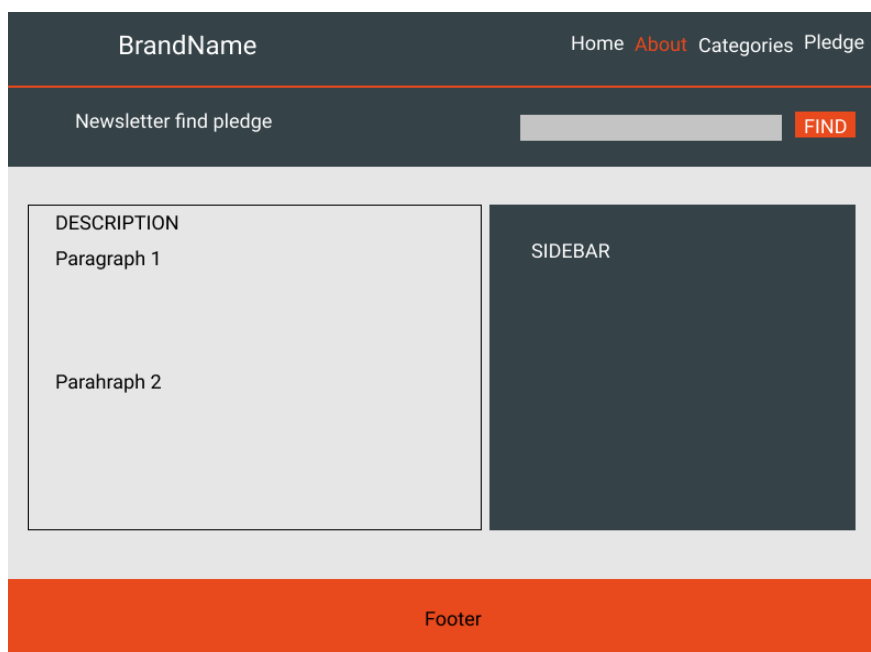


Рисунок 3.3 – Макет сайта, страница информации. SIDEBAR – боковая панель, ссылка на заполнение формы.

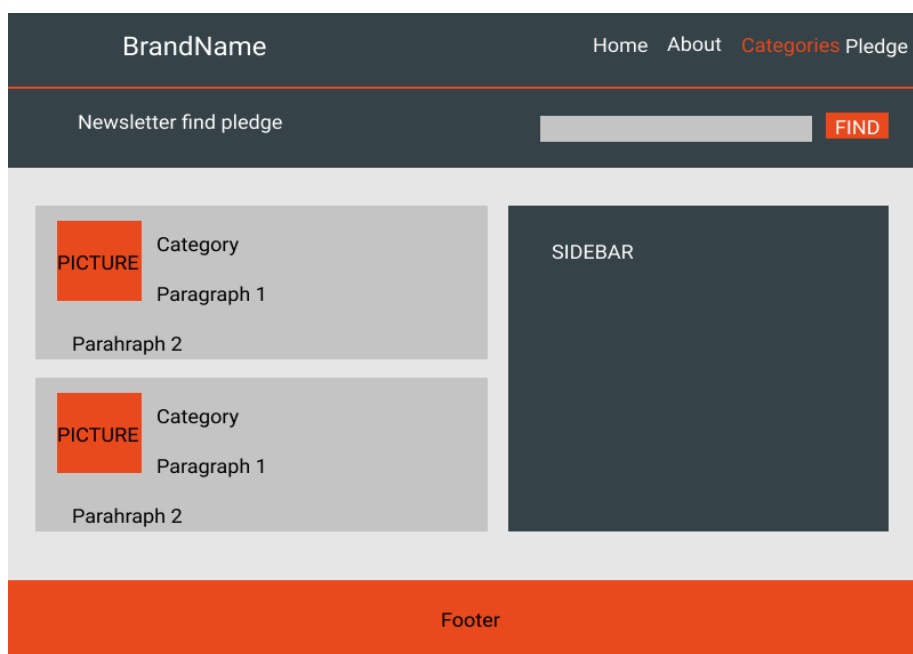


Рисунок 3.4 – Макет сайта, страница категорий. Category – конкретная категория, PICTURE – фото украшения.

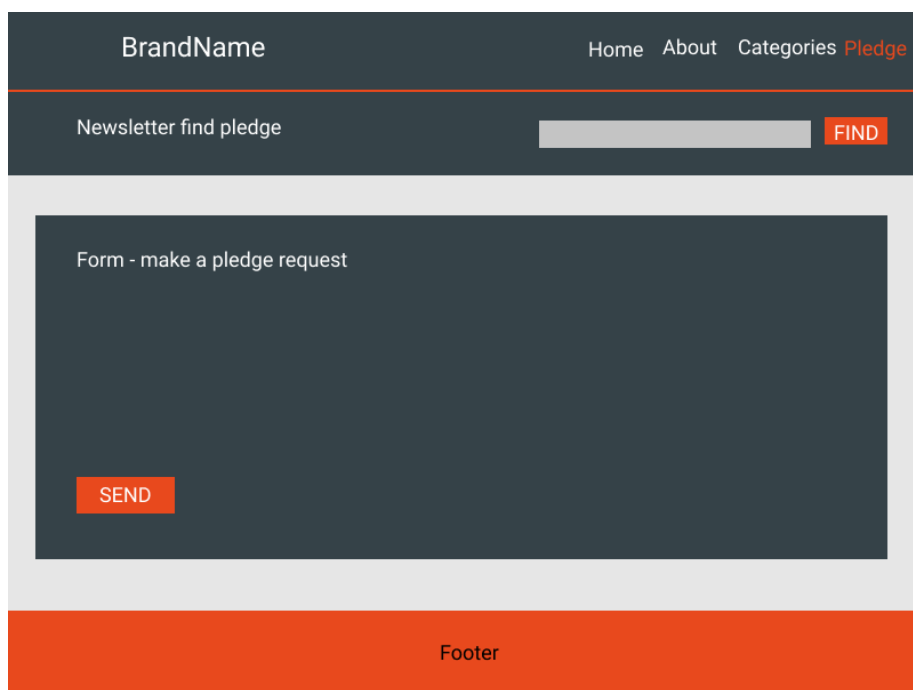


Рисунок 3.5 – Макет сайта, страница составления заявки. Form – заполнение полей для подачи заявки на оценку.

## 4. Иерархическая архитектура модулей

### Диаграмма потоков данных DFD

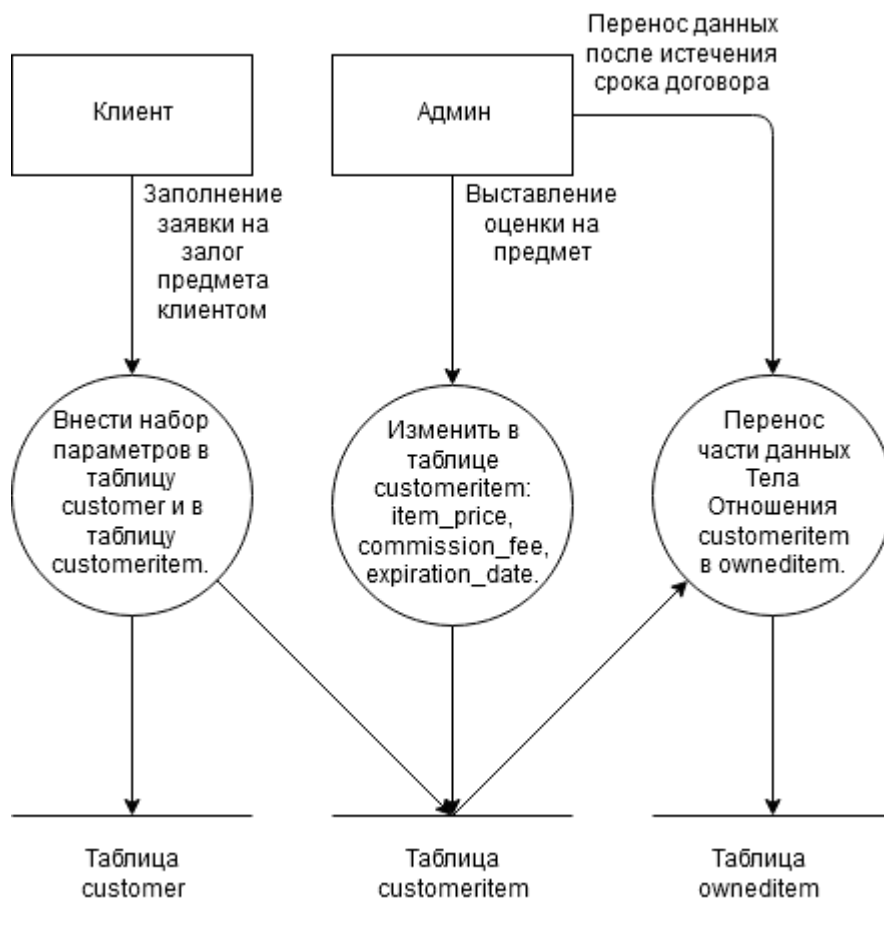


Рисунок 4.1 - Диаграмма потоков данных (нотация Йордона – Де Марко).

В дальнейшем для определения сущностей и их атрибутов будем использовать обозначение вида:

Клиенты ломбарда:

**Клиент** (паспорт серия и номер, имя, фамилия, отчество, почта)

**Ключ сущности:** паспорт серия и номер

Предметы, заложенные клиентами:



**Карта Клиента** (Индивидуальный номер (id) предмета, Категория, Название предмета, Краткое описание, Цена за предмет, Комиссионные, Срок возврата денег, Паспорт владельца, Фотография предмета)

**Ключ сущности:** Индивидуальный номер (id) предмета

Предметы ломбарда:

**Карта Ломбарда** (Индивидуальный номер (id) предмета, Категория, Название предмета, Краткое описание, Фотография предмета, Изначальная цена, Нынешняя цена)

**Ключ сущности:** Индивидуальный номер (id) предмета

**Связи между сущностями «Сущность-связь»**

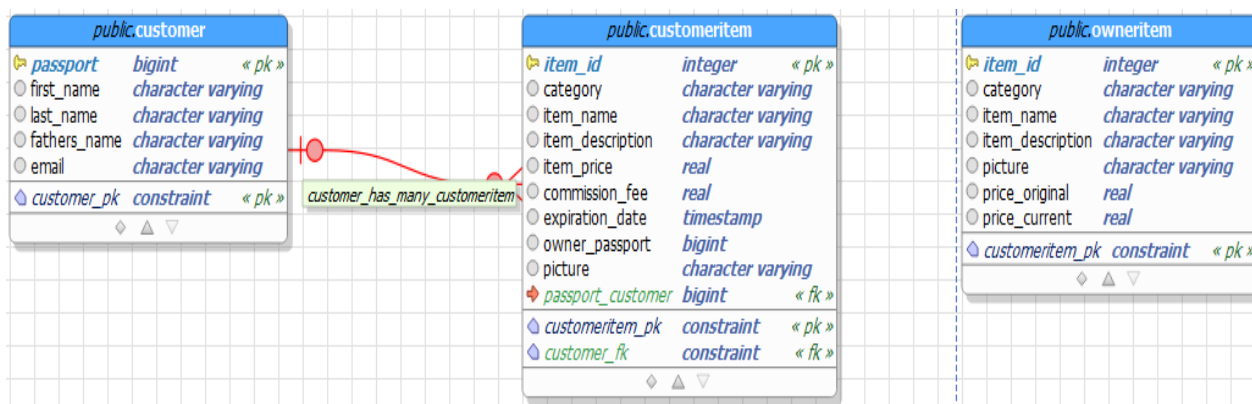


Рисунок 4.2 – Диаграмма отношений «Сущность-Связь» (ER диаграмма).

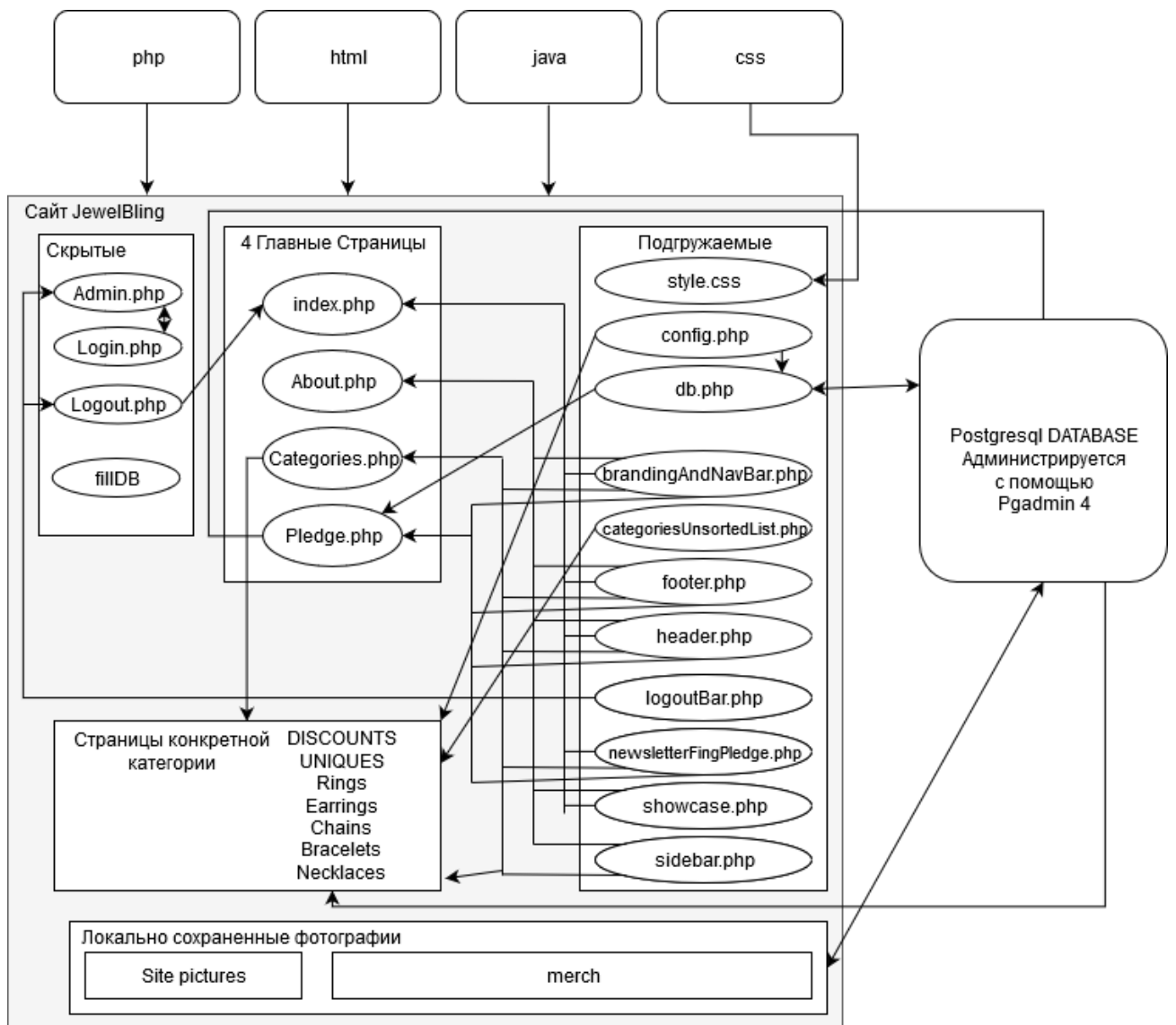


Рисунок 4.3 – Иерархическая архитектура всех модулей сайта.

## 5. База данных

Подробное словесное описание объектов предметной области и реальных связей, которые присутствуют между ними, уже описаны в предыдущих темах. Известно, какие таблицы и какие данные должны быть в базе данных. Необходимо осуществить нормализацию отношений.

### Нормализация отношений, построение нормальных форм

#### Отношение **customer**:

Атрибут отношения	Пример
#passport	7616123456
first_name	Vladimir
last_name	Demidov
fathers_name	Yuryevich
email	vladimirdemidov19@gmail.com

Таблица 5.1 – Отношение клиент с примером данных.

- Переменные отношения **customer** находятся в 1 нормальной форме (1NF), так как все атрибуты отношения атомарны. (не обеспечивается явная видимость внутренней структуры атрибутов, со всеми значениями можно обращаться только с помощью операций, определенных в соответствующем типе данных).

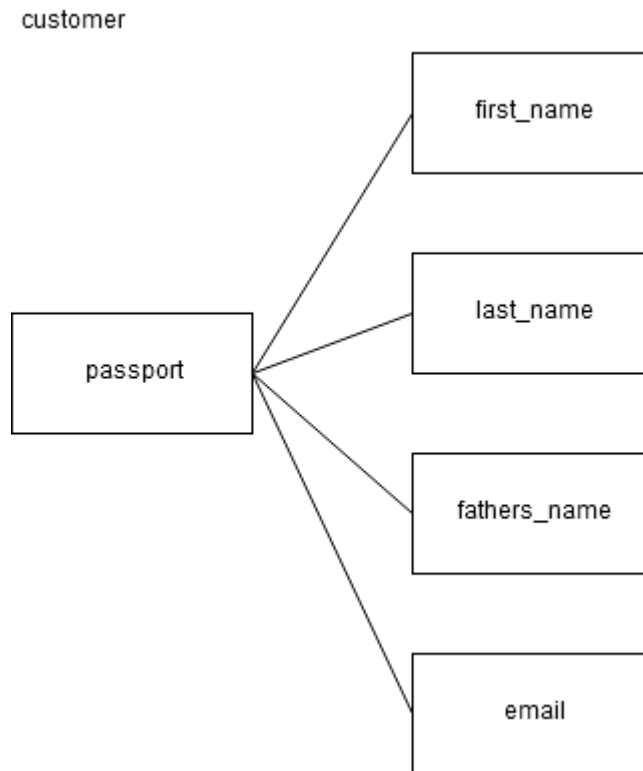


Рисунок 5.1 – Схема функциональных отношений (FD) customer.

- Переменные отношения **customer** находятся во 2 нормальной форме (2NF), так как они уже находятся в 1 нормальной форме (1NF), и каждый неключевой атрибут своего отношения минимально функционально зависит от первичного ключа в своем отношении.

#### Отношение customeritem:

Атрибут отношения	Пример
#item_id	1
category	RINGS
item_name	moonring
item_description	silver ring with moonstone
item_price	2999

commission_fee	149.95
expiration_date	21.02.20
owner_passport	7616123456
picture	myPrecious.jpg

Таблица 5.2 – Отношение предметы клиента с примером данных.

- Переменные отношения **customeritem** находятся в 1 нормальной форме (1NF), так как все атрибуты отношения атомарны. (не обеспечивается явная видимость внутренней структуры атрибутов, со всеми значениями можно обращаться только с помощью операций, определенных в соответствующем типе данных).

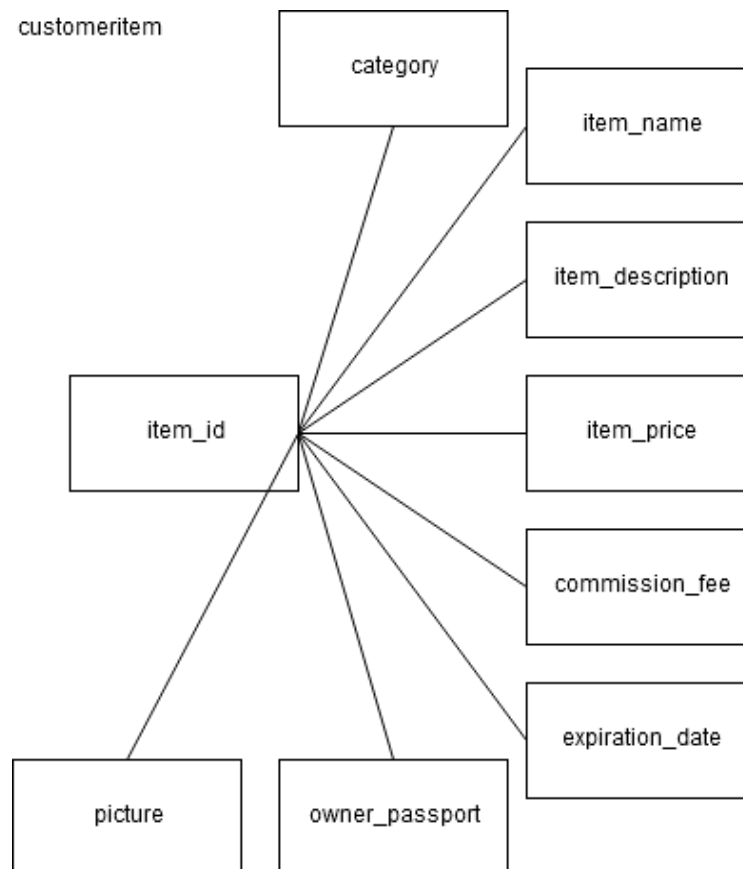


Рисунок 5.2 - Схема функциональных отношений (FD) customeritem.

- Переменные отношения **customeritem** находятся во 2 нормальной форме (2NF), так как они уже находятся в 1 нормальной форме (1NF), и каждый

неключевой атрибут своего отношения минимально функционально зависит от первичного ключа в своем отношении.

**Отношение owneditem:**

Атрибут отношения	Пример
#item_id	2
category	UNIQUE
item_name	Golden Crown
item_description	925 probe golden crown.
picture	a_crown_for_a_king.png
price_original	29999
price_current	28499.05

Таблица 5.3 – Отношение предметы в собственности ломбарда с примером данных.

- Переменные отношения **owneditem** находятся в 1 нормальной форме (1NF), так как все атрибуты отношений атомарны. (не обеспечивается явная видимость внутренней структуры атрибутов, со всеми значениями можно обращаться только с помощью операций, определенных в соответствующем типе данных).

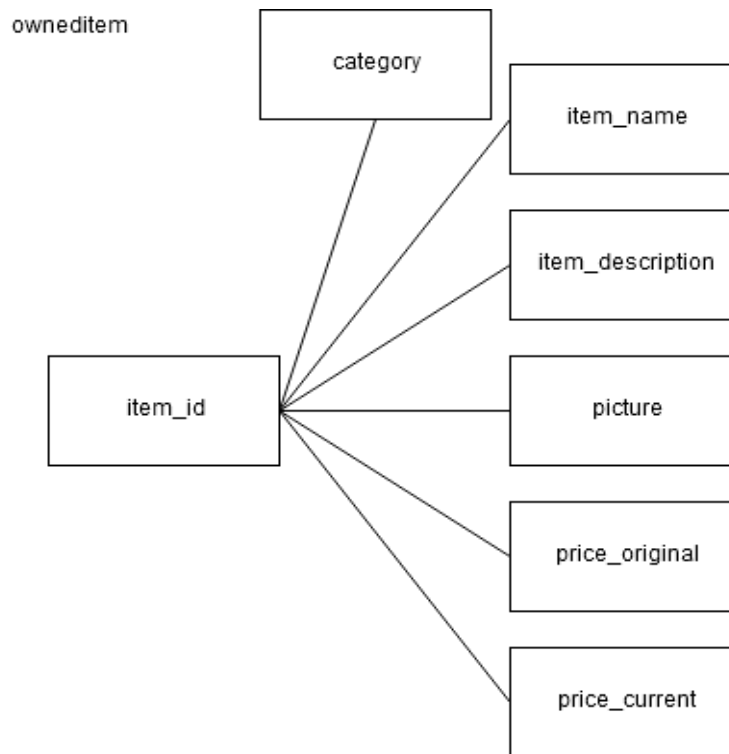


Рисунок 5.3 - Схема функциональных отношений (FD) owneditem.

- Переменные отношения **owneditem** находятся во 2 нормальной форме (2NF), так как они уже находятся в 1 нормальной форме (1NF), и каждый неключевой атрибут своего отношения минимально функционально зависит от первичного ключа в своем отношении.
- Переменные отношений **customer**, **customeritem**, **owneditem** находятся в 3 нормальной форме (3NF), так как они находятся во второй нормальной форме, и каждый неключевой атрибут своего отношения нетранзитивно функционально зависит от первичного ключа своего отношения (не существует такого множества атрибутов, чтобы одно из отношений определяло его, и чтобы одновременно оно определяло другое отношение).
- Переменные отношений **customer**, **customeritem**, **owneditem** находятся в Нормальной форме Бойса-Кодда (BCNF), так как детерминанты всех их функциональных зависимостей являются потенциальными ключами.
- Переменные отношений **customer**, **customeritem**, **owneditem** находятся в 4 Нормальной форме (4NF), так как они уже находятся в нормальной

форме Бойса-Кодда, и все нетривиальные многозначные зависимости фактически являются функциональными зависимостями от их потенциальных ключей.

Когда человек оставляет заявку на оценку украшения, для него создается поле в отношении customer. Один человек может создавать множество заявок, но поле customer будет только одно. Ключ отношения customer – passport, является единственным и индивидуальным для каждого человека. На один атрибут passport только один экземпляр first\_name, last\_name, fathers\_name и email.

У каждого поля в отношении customeritem первичный ключ – item\_id индивидуален для каждого предмета – можно добавлять неограниченное количество предметов. В каждом поле есть атрибут owner\_passport, с помощью которого можно узнать какой клиент добавил предмет. Для первичного ключа item\_id существует единственный экземпляр атрибутов category, item\_name, item\_description, item\_price, owner\_passport, picture, заполняемых клиентом, и единственный экземпляр атрибутов item\_price, commission\_fee, expiration\_date заполняемых администратором (оценка предмета). Между первичными ключами passport отношения customer и item\_id отношения customeritem может быть функциональная зависимость.

При переходе предмета в собственность ломбарда происходит перенос поля из отношения customeritem в поле owneditem с первичным ключом item\_id, для которого существуют в единственном экземпляре атрибуты item\_name, item\_description, picture, price\_original, price\_current.

## **Реализация декларативных ограничений целостности и проектирование таблицы Базы Данных**

Ограничение целостности – некоторое утверждение, которое может быть либо истинным, либо ложным, в зависимости от состояния базы данных. База данных находится в согласованном (целостном) состоянии, если выполнены все ограничения целостности, определенные для базы данных. Любое ограничение целостности появляется как следствие определенных свойств объектов предметной области и/или их взаимосвязей.



У всех клиентов должен быть свой паспорт, по которому в дальнейшем можно будет вести поиск предметов, которые они сдают в ломбард. Так как паспорт для каждого человека индивидуален, он должен быть первичным ключом. У всех клиентов также должны быть имя, фамилия и отчество. У всех клиентов должна быть почта, чтобы можно было их проинформировать о выставлении оценки на их предметы.

#### **Ограничения целостности для Отношения customer:**

- Атрибут passport – not null;
- Атрибут passport – primary key;
- Атрибут first\_name – not null;
- Атрибут last\_name – not null;
- Атрибут fathers\_name – not null;
- Атрибут email – not null;

Все предметы клиентов должны иметь собственный идентификационный номер. У каждого предмета должна быть категория, для сортировки предметов (при неправильном указании категории поле этого атрибута редактируется администратором при оценке предмета, и на поиск это не влияет). Предмету необходимо дать название. Описание для предмета – не обязательно, но желательно. Цена предмета – выставляется администратором, и не должна быть меньше нуля. Комиссионные – выставляются администратором, и не должны быть меньше нуля. Дата истечения срока договора – выставляется администратором, и она должна быть не равной текущему дню. Паспорт владельца должен быть заполнен. Фотография предмета тоже необходима для качественной оценки.

#### **Ограничения целостности для Отношения customeritem:**

- Атрибут item\_id – not null;
- Атрибут item\_id – default self-increment +1;
- Атрибут item\_id – primary key;

- Атрибут category – not null;
- Атрибут item\_name – not null;
- Атрибут item\_price – check greater than 0;
- Атрибут commission\_fee – check greater than 0;
- Атрибут expiration\_date – check current\_timeStamp != today;
- Атрибут owner\_passport – not null;
- Атрибут owner\_passport – foreign key;
- Атрибут picture – not null;

Все предметы, находящиеся в собственности ломбарда, должны иметь собственный идентификационный номер. Они должны иметь категорию, чтобы отображаться на сайте в соответствующих категориях. Они должны иметь название. Описание – желательно. Клиентов не получится заинтересовать товаром без фотографии – она обязательна. Обязательным должен быть атрибут изначальной цены для предмета, так как скидки будут вычисляться из этого значения в обязательный атрибут нынешней цены. При этом изначальная цена и нынешняя цена должны быть больше нуля.

#### **Ограничения целостности для Отношения owneditem:**

- Атрибут item\_id – not null;
- Атрибут item\_id – default self-increment +1;
- Атрибут item\_id – primary key;
- Атрибут category – not null;
- Атрибут item\_name – not null;
- Атрибут picture – not null;
- Атрибут price\_original – not null;
- Атрибут price\_original – check greater than 0;

- Атрибут price\_current – not null;
- Атрибут price\_current –check greater than 0;

### Связи между сущностями «Сущность-связь»

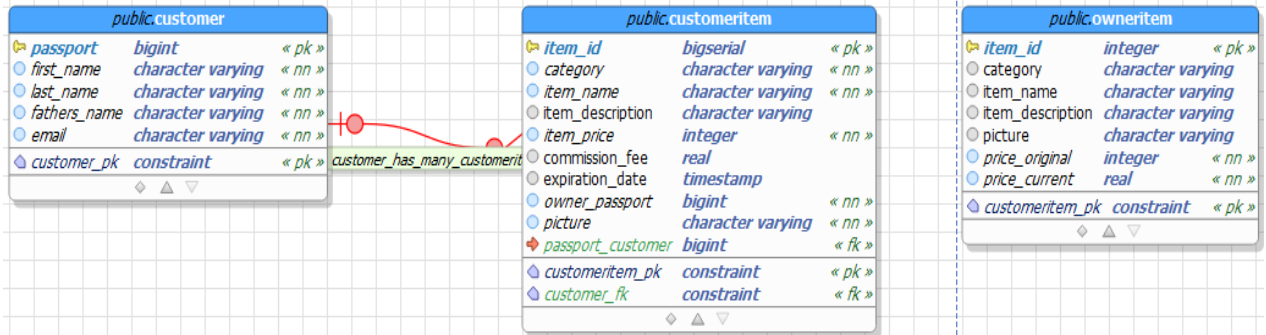


Рисунок 5.4 – Диаграмма отношений «Сущность-Связь» базы данных в целостном состоянии (ER диаграмма).

## **6. Программная реализация**

Подробное описание алгоритма работы каждого файла. Для полноценного и быстрого создания веб страниц были созданы дополнительные .php файлы, которые необходимо подгрузить, чтобы они отображались на сайте.

### **Четыре главные страницы:**

#### **Домашняя страница index.php**

В заголовочной части сайта – подгрузка общего заголовка.

В содержательной части сайта – подгрузка брэндовой фотографии и описания, панели навигации, поиск заявок. Затем – контейнерные части, содержащие три самых посещаемых категории сайта – скидки, уникальные, и все категории. При наведении на контейнер он подсвечивается, при нажатии – отправляет по ссылке в соответствующую ему категорию. В конце содержательной части – подгрузка завершительной части.

#### **Страница описания about.php**

В заголовочной части сайта – подгрузка общего заголовка.

В содержательной части сайта – подгрузка панели навигации, поиск заявок. Затем – контейнер с описанием назначения сайта. В конце контейнера – подгрузка боковой панели на оставление заявки. В конце содержательной части – подгрузка завершительной части.

#### **Страница категорий categories.php**

В содержательной части сайта – подгрузка панели навигации, поиск заявок. Затем – контейнер, в котором несортированный список с категориями и их описаниями. Наведение на элемент списка подсвечивает его, а нажатие – переводит на второстепенную страницу с конкретной категорией. В конце контейнера – подгрузка боковой панели на оставление заявки. В конце содержательной части – подгрузка завершительной части.

#### **Страница заполнения заявки на залог товара pledge.php**

В этом файле два блока. В первом – php код, в него подгружается конфигурационный файл, в котором находится вся информация о базе данных. Затем отлавливается нажатие клавиши «отправить заявку» после заполнения всех полей (кнопка не сработает при отсутствии необходимой информации в полях). Передача заполненных данных в переменные.

Затем необходимо проверить, есть ли клиент в базе данных, или это новый клиент (проверка происходит посредством SQL вставки, в которой идет поиск по паспорту). Если клиента нет – создается информация о нем. Если есть – этот шаг пропускается. Затем в базу данных добавляется информация о предмете. При ошибке обновления базы данных выводится сообщение с помощью скрипта на языке java.

Во втором блоке – подгрузка панели навигации, поиска по паспорту, затем контейнер с полями разных типов. В конце – закрывающая часть.

## **Дополнительные страницы:**

### **Страницы конкретных категорий**

Каждая страница для каждой категории абсолютно идентична, однако их написание происходит быстро из-за подгрузки всех необходимых частей.

В этом файле два блока. В первом блоке – подгрузка файла с информацией о базе данных, затем создание очереди SQL вставкой, где происходит выбор (SELECT) всех полей таблицы, в которых категория соответствует нужной. В данном случае категория – DISCOUNTS. Эта очередь преобразовывается в ассоциативный массив, в котором названия полей легко найти по названиям атрибутов в самой базе, и освобождается переменная очереди, прерывается связь с базой.

Во втором блоке происходит подгрузка панели навигации, поиска, затем подгрузка файла в котором описан цикл перебора ассоциативного массива и его вывод несортированным списком на сайте, затем закрывающая часть.

Другие категории выглядят аналогично.

## **Страница результатов поиска – findPledge.php**

В файле два блока. В первом блоке – подключение к файлу с информацией о базе данных, поиск в базе по паспорту с помощью SQL вставки.

Во втором блоке происходит подгрузка панели навигации, поиска, затем подгрузка файла в котором описан цикл перебора ассоциативного массива при совпадающих значениях паспорта, и его вывод несортированным списком на сайте, затем закрывающая часть.

## **Скрытая страница для администратора – admin.php**

Если неавторизованный пользователь переходит на страницу admin.php его пересылают на страницу login.php, в которой два блока. В первом блоке создается переменная сессии, и булево значение наличия или отсутствия авторизации. Это значение можно переносить на другие страницы, и с помощью php вставок сделать проверки наличия авторизации и добавление возможностей.

Во втором блоке – форма, где необходимо ввести имя пользователя и пароль. Оба параметра берутся из главного файла, где хранится информация о базе данных – имя супер пользователя и пароль. Только администратор базы данных сможет авторизоваться.

Проверка переменной наличия авторизации происходит на странице с заглушками для редактирования базы данных. Если авторизация есть – загружается панель и кнопка выхода. Если нет – ее не видно вообще. После авторизации появляются кнопки – заглушки.

В файле admin.php происходит подгрузка необходимых файлов, и три кнопки – заглушки.

## **Страница авторизации – login.php**

При нажатии на кнопку выхода из системы, происходит переход на скрытую страницу logout.php, в которой обнуляется переменная наличия авторизации. затем происходит переадресация на главную страницу.

## **Страница выхода из системы – logout.php**

Обнуление переменной наличия авторизации, и перенос на главную.

## **Конфигурационные файлы**

### **Файл данных о базе – config.php**

В этом файле хранятся необходимые данные для доступа к базе данных, а также они используются в виде глобальных переменных для авторизации администратора.

### **Файл создания связи с базой – db.php**

В этом файле устанавливается связь с базой данных с помощью информации из конфигурационного файла.

### **Файл описания внешнего вида страницы style.css**

Единственный файл, где описывается внешний вид сайта. Определение нужного вида происходит с помощью разных id, которые можно использовать в других файлах.

В файле код разделен на блоки с названиями, какой блок за что отвечает. Первый блок объявляет глобальные параметры – цвета. Блок содержательной части – общий для всех сайтов. Все контейнеры тоже описаны в глобальной части. Затем идут элементы несортированных списков. После него заголовочная часть, блок фотографии бренда, поисковый блок, все содержательные квадраты, боковая панель, главная панель содержательной части, услуги (предоставление товаров – категории), закрывающая часть. В конце файла описываются размеры для планшетного разрешения экрана. Этот блок меняет вид сайта в зависимости от размеров окна.

## **Подгружаемые файлы**

### **Файл брендинга и навигационной панели – brandingAndNavBar.php**

В этом файле отображается навигационная панель и название бренда «JewelBling». В содержательной части – php вставки с условными операторами. С их помощью задается подсветка той страницы, на которой сейчас находится пользователь.

### **Файл категорий, несортированный список – categoriesUnsortedList.php**

Этот файл загружается в каждой категории. В первом блоке идет определение, какую именно категорию необходимо загрузить из базы данных. Объявлена функция для вывода скидки, но эта функция используется только в категории скидочных товаров, посредством условного оператора. Во втором блоке – цикл перебора ассоциативного массива и вывод результатов.

### **Файл, завершающий содержание сайта – footer.php**

В файле выводится параграф информации.

### **Заголовочный файл – header.php**

В этом файле идет объявление всех необходимых параметров для работы сайта, загрузка шрифтов, определение названия в закладке.

### **Файл панели с кнопкой выхода из системы для админа – logoutBar.php**

В файле – контейнер с кнопкой для выхода из системы.

### **Файл панели поиска заявок на залог – newsletterFindPledge.php**

В первом блоке происходит передача переменной с паролем в сессию, затем во втором блоке – поля для поиска по паспорту. При нажатии на поиск первый блок отлавливает значение поля паспорт, и отправляет их в сессию.

### **Файл описания ломбарда – showcase.php**

В файле – параграф с описанием деятельности ломбарда.

### **Файл боковой панели – sidebar.php**

В файле – параграф и кнопка, отправляющая на другую страницу.



## **7. руководство пользователя**

### **Область применения**

Сайт «JewelBling» предназначен для предоставления посетителям списка товаров, имеющихся в наличии на данный момент, а также предоставления возможности узнать сколько стоит украшение, которое посетители хотели бы отдать в залог, без необходимости личного посещения ломбарда.

### **Краткое описание возможностей**

Основными целями создания сайта «JewelBling» являются:

- Показ товаров посетителям сайта;
- Оценка украшений с помощью фотографий;
- Проверка статуса оценки.

Для реализации целей приведен перечень возможностей:

- Возможность заполнения бланка и отправки украшения на оценку;
- Возможность поиска заявок на оценку по паспорту;
- Возможность просмотра каталога товаров ломбарда;
- Ведение записей всех клиентов, отправивших заявки на оценку товаров;
- Ведение записей всех предметов, которые были отправлены на оценку;
- Ведение записей всех предметов в собственности ломбарда;
- Добавление, изменение, удаление этой информации в базе данных.

### **Уровень подготовки пользователя**

Пользователи (посетители) сайта должны иметь базовые навыки владения компьютером, и должны хорошо ориентироваться в браузере.

### **Назначения и условия применения**

Сайт «JewelBling» предназначен для:

- Представление посетителям списка товаров в наличии;
- Возможность оценки украшений в режиме онлайн;
- Поиск своих заявок по паспорту для проверки их статуса.

При отправке заполненного бланка с личными данными, и фотографией, происходит добавление информации в базу данных: добавляется новая строка в поле клиент, при условии что его паспорт уже не существует в таблице, добавляется новая строка в таблице предмет клиента.

### **Виды деятельности, функции**

Сайт «JewelBling» предназначен для следующих видов деятельности:

- Просмотр товаров по категориям;
- Оценка украшений в режиме онлайн;
- Просмотр статуса заявки на оценку.

### **Программные и аппаратные требования к системе**

Для нормальной работы сайта необходимо выполнение следующих условий:

- Операционная система Windows XP/Vista/7/8/10 или Mac OS Tiger/Leopard/Snow Leopard;
- Браузер Google Chrome 14 и выше, Mozilla Firefox 3.6 и выше, Opera 10 и выше, Safari 5 и выше, IE 8.0 и выше.

### **Подготовка к работе**

Для того чтобы перейти на сайт «JewelBling» необходимо, чтобы программные и аппаратные характеристики системы соответствовали требуемым, а также наличие подключения к интернету. Чтобы перейти на главную страницу сайта, необходимо ввести соответствующий запрос в любом из поисковых систем. Сайт реагирует на следующие ключевые слова: **web-design, html, css,** или можно перейти по ссылке <http://localhost/phpsandbox/JewelBling>

## **Состав дистрибутива (для администрации)**

Для редактирования таблиц в базе данных необходимо использовать PgAdmin 4 или выше.

## **Запуск системы (для администрации)**

Для полноценной работы сайта необходимо установить XAMPP 7.4.1 или выше. Необходимо открыть файл хampp-control.exe в папку, куда была произведена установка, затем запустить сервер. Для просмотра данных в таблицах напрямую необходимо открыть PgAdmin 4 или выше, и войти по данным суперпользователя.

## **Проверка работоспособности системы**

При запуске PgAdmin и хampp-control не должны быть ошибки. Если после запуска сервера через хampp-control, при переходе на сайт происходит успешное отображение сайта, значит сайт готов к использованию.

## **Описание операций**

У сайта JewelBling существует четыре главные страницы.

- **index.php**
- **about.php**
- **categories.php**
- **pledge.php**

**index.php** – главная страница сайта. На ней есть основная информация, ссылки на популярные категории товаров, лаконичное описание предприятия.

**about.php** – страница с информацией о предприятии, законы, соглашения, любой информационный длинный текст должен быть именно в этой части.

**categories.php** – страница с категориями товаров ломбарда. Отсюда посетители страницы могут просмотреть товары ломбарда по категориям. Все товары (названия, описания, цены, фотографии) загружены с базы данных postgresql.

**pledge.php** – страница для заявок на оценку украшений. Необходимо заполнить форму, загрузить фотографию украшения для оценщиков.

Помимо главных страниц есть и другие, которые тоже составляют важную часть функционала сайта.

- Своя страница для каждой категории;
- Результаты поиска заявок по паспорту;
- Скрытая страница авторизации для администратора;
- Загрузка скрытой страницы редактирования базы данных через сайт.

Когда посетитель переходит в любую категорию украшений, происходит переход на другую страницу, в которой отображаются все товары ломбарда в данной категории. Загрузка предметов происходит из базы данных.

Каждый пользователь, оставивший заявку на оценку украшения, может в любой момент просмотреть статус оценки. Для этого, на всех страницах сайта присутствует элемент для поиска заявки по паспорту. После его заполнения происходит переход на новую страницу, на которой выводятся результаты поиска. В зависимости от результата, будут выведены блоки с украшениями, ожидающими оценку или имеющие оценку, или же будет сказано что поиск ничего не дал.

Пользователи могут оставлять заявки и смотреть результат оценки, вне необходимости регистрации на сайте. Однако в любом сайте должна быть опция для авторизации администратора. Для этого существует скрытая страница /admin.php, на которую можно пройти только зная ссылку. При переходе по ссылке сразу происходит переход на страницу авторизации, и только затем открывается доступ к правам администратора.

## ЗАКЛЮЧЕНИЕ

Проектирование сайта, пользовательского интерфейса и пользовательского опыта включает в себя пять уровней. На стратегическом уровне необходимо наиболее точно определить общее описание и назначение сайта – демонстрация товаров, онлайн оценка украшений. На уровне возможностей необходимо составить список функциональных возможностей сайта – сайт, на котором пользователи смогут просматривать каталог товаров, и отправлять на оценку украшения. На этом уровне составляется Use-case диаграмма. На уровне структуры описывается способ реализации функционала, а также составляется концептуальная модель сайта. На уровне компоновки определяется интерфейс пользователя, расположение основных элементов и их группировка. Была составлена модель сайта с главными страницами. На уровне поверхности реализуется то, что пользователь должен видеть – тексты, иконки, кнопки. При этом, проектирование каждого следующего уровня начинается до окончания предыдущего.

В курсовом проекте были рассмотрены основные методы разработки веб-приложения – авторизация, передача переменных с помощью сессии, подключение и взаимодействие с базой данных postgresql, и были освоены основные принципы создания отзывчивого, качественного и понятного веб-интерфейса.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Habr. Веб разработка. С чего начать? [Электронный ресурс]//Электронный Текстовый документ. Режим доступа:  
<https://habr.com/ru/post/357720/> свободный
2. Wikipedia. Веб разработка [Электронный ресурс]//Электронный Текстовый документ. Режим доступа:  
<https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0> свободный;
3. Professor web. Основы веб программирования [Электронный ресурс]//Электронный Текстовый документ. Режим доступа:  
[https://professorweb.ru/my/html/html5/level1/web\\_info.php](https://professorweb.ru/my/html/html5/level1/web_info.php) свободный;
4. astracms. Архитектурные особенности и терминология программирования [Электронный ресурс]//Электронный Текстовый документ. Режим доступа: <http://wiki.astracms.ru/architecture> свободный;
5. ktso. Нормативная документация, гост функциональная спецификация программирования [Электронный ресурс]//Электронный Текстовый документ. Режим доступа: <http://www.ktso.ru/p-iso-mek-18045/gost-p-iso-mek-18045-2008/gost-p-iso-mek-18045-2008-10-6.php> свободный;
6. rugost. Документ «Руководство пользователя» [Электронный ресурс]//Электронный Текстовый документ. Режим доступа:  
[http://www.rugost.com/index.php?option=com\\_content&view=article&id=181:q-q42&catid=27&Itemid=64](http://www.rugost.com/index.php?option=com_content&view=article&id=181:q-q42&catid=27&Itemid=64) свободный;

7. Netology. Пять уровней UX [Электронный ресурс]//Электронный Текстовый документ. Режим доступа: <https://netology.ru/blog/5level-ux> свободный.