КУРСОВАЯ  РАБОТА

по дисциплине: «Разработка Интернет-приложений»

Тема:  Интернет-магазин

Оглавление

[1.Цель и задачи курсовой работы 2](#__RefHeading__64_441018704)

[2. Проектирование. 3](#__RefHeading__66_441018704)

[2.1. Проектирование логической структуры работы 3](#__RefHeading__68_441018704)

[2.2. Выделение основных групп функций приложения. 3](#__RefHeading__70_441018704)

[2.3. Формирование перечня функций для каждой группы 4](#__RefHeading__72_441018704)

[2.4. Разработка структуры хранения данных проекта 15](#__RefHeading__74_441018704)

[2.5. Разработка структуры классов для обработки данных. 17](#__RefHeading__76_441018704)

[2.6. Формирование перечня интерфейсов для ввода данных и определение заполняемых полей в каждом интерфейсе 24](#__RefHeading__78_441018704)

[2.7. Создание статических страниц для каждого раздела 25](#__RefHeading__80_441018704)

[2.8. Проектирование карты переходов 29](#__RefHeading__82_441018704)

[3. Создание базы данных 31](#__RefHeading__84_441018704)

[4. Разработка 35](#__RefHeading__86_441018704)

[5. Заключение 35](#__RefHeading__2160_2004558005)

[6. Литература 36](#__RefHeading__2162_2004558005)

# 1.Цель и задачи курсовой работы

Цель — создать интернет-магазин.

Задачи, которые должно решать программное обеспечение интернет-магазина можно разделить на 3 части. Первая часть относится к возможностям администрирования интернет-магазина, вторая относится к возможностям, которые реализует программное обеспечение для продавца, а третье - для покупателей.

Задачи, которые программное обеспечение позволяет выполнять администратору:

- просматривать каталог товаров;

- добавлять новый товар;

- редактировать данные о товаре;

- удалять товар;

Задачи, которые программное обеспечение позволяет выполнять продавцу:

- просматривать список заказов;

- удалять заказы;

- просматривать каталог товаров (с ценой товара);

- оформлять продажу товара.

Задачи, которые программное обеспечение позволяет выполнять покупателю:

- просматривать каталог товаров (с ценой товара);

- добавлять товар в корзину;

- просматривать, редактировать и печатать содержимое корзины;

- регистрация покупателя.

Для реализации интернет-магазина разработчик принял решение использовать альтернативный вариант технологий, приведенный в задании для курсового проекта: сервер баз данных MySQL. В качестве языка программирования используется PHP.

# 2. Проектирование.

## 2.1. Проектирование логической структуры работы

Для проектирования логической структуры интернет-магазина необходимо выделить основные объекты предметной области и связи между ними.

Рассмотрим процесс функционирования интернет-магазина:

Администратор вызывает страницу администрирования магазина. Ему предлагается ввести логин и пароль. После успешного ввода пароля администратору открывается каталог товаров. Администратору становятся доступны функции добавления нового товара, редактирования товара и удаления товара. Товар характеризуется: названием, ценой и количеством на складе. Кроме добавления товара администратору предлагается функция поиска товара по названию. При успешном поиске администратор также может изменить найденный товар или удалить его.

Продавец вызывает свою рабочую страницу магазина. Ему предлагается ввести логин и пароль. После успешного ввода пароля продавец получает страницу с непогашенными заказами. Заказ содержит дату заказа, название товара, количество товара, стоимость заказа, данные покупателя: ФИО. Продавец может удалить заказ и оформить продажу, при этом распечатывается счет на оплату, а в каталоге товаров изменяется количество товара.

Покупатель, заходя на главную страницу магазина, может просматривать каталог товаров, производить по нему поиск и добавлять в корзину понравившийся товар, также просматривать и редактировать содержимое корзины. Чтобы товары из корзины стали заказом покупателю необходимо зарегистрироваться. При этом покупатель вводит обязательную информацию о себе: ФИО, телефон и адрес доставки. После этого пользователь может оформить заказ.

Таким образом, анализируя функционирование интернет-магазина можно выделить следующие объекты предметной области:

-*Пользователи интернет-магазина* (администратор, продавец и покупатель). В зависимости от типа пользователя — доступны те или иные ресурсы;

- *Каталог товаров* состоит из *товаров;*

-*Товар;*

*-Список заказов* состоит из *заказов;*

*-Заказ* объединяет информацию о *товаре* и *покупателе;*

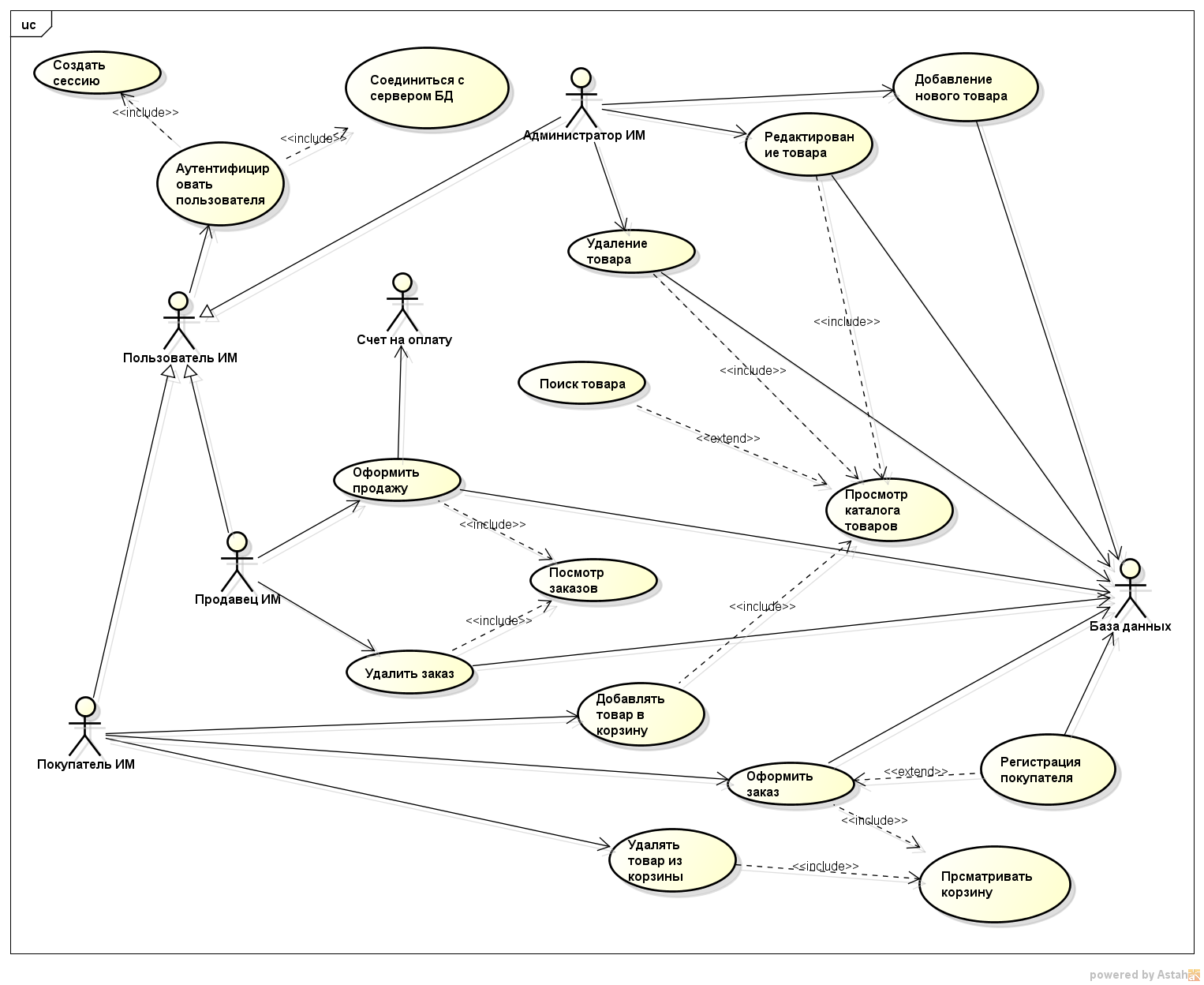
*- Счет на оплату;*

*- Покупатель;*

*-Корзина* состоит из *товаров.*

## 2.2. Выделение основных групп функций приложения.

Для выделения основных групп функций воспользуемся диаграммой вариантов использования [5].



## 2.3. Формирование перечня функций для каждой группы

Реализуем описание функций с помощью описания потоков событий и диаграмм последовательностей [5].

1. Аутентифицировать пользователя

1.1. Краткое описание. Для того, чтобы войти на сайт магазина пользователь должен быть опознан магазином, подтвердить свой статус паролем и получить соответствующие права на использование сайта магазина.

1.2. Предусловия. Администратор и Продавец заходят на сайт магазина с использованием страницы аутентификации form\_auth.php. Покупатель при входе на сайт fokinshop/index.php автоматически получает пользовательские имя пользователя и пароль.

1.3. Основной поток событий.

1.3.1.Пользователь вводит в браузере адрес страницы магазина, в зависимости от своего статуса.

1.3.2. Сайт предоставляет страницу, на которой есть поля ввода имени пользователя и пароля.

1.3.3. Пользователь вводит имя пользователя и пароль и нажимает кнопку «Войти».

1.3.4. Браузер отправляет post запрос на сервер, где в таблице базы данных хранятся подлинные значения.

1.3.5. Сервер проверяет совпадают ли имя пользователя и пароль с истинными и при условии совпадения открывает сессию с переменными сессии, которые являются именем пользователя и паролем на подключение к базе данных, в которой хранится сайт.

Имя пользователя и пароль (серверные) разграничивают доступ к таблицам базы данных:

Администратор имеет доступ на чтение и редактирование для таблицы товаров;

Продавец имеет доступ на чтение, редактирование и удаление для таблицы заказов;

Покупатель имеет доступ на чтение к таблице товаров, на добавление данных в таблицу покупателей, на добавление данных в таблицу заказов.

1.3.6. После успешной аутентификации происходит соединение переход на запрашиваемую страницу. Или альтернативный вариант.

1.4. Альтернативный вариант событий 1.

Покупатель заходит на сайт - покупателю сразу предоставляются права доступа, описанные в пункте 1.3.5.

1.5.Поток ошибок.

1.5.1. В случае, если имя пользователя и пароль ошибочные — выдается соответствующее сообщение.

2. Добавление нового товара.

2.1. Краткое описание. Администратору предоставляется возможность добавления нового товара.

2.2. Предусловия. Успешная аутентификация.

2.3. Основной поток событий.

2.3.1.Администратор выбирает команду «Добавить новый товар»

2.3.2. Сервер предоставляет администратору форму для ввода информации о новом товаре:

название товара, цена и количество единиц.

2.3.3. После ввода данных администратор нажимает кнопку «Добавить» и товар добавляется в таблицу «Товары» базы данных. Или альтернативный вариант.

2.4. Альтернативный вариант событий 1.

Администратор нажимает кнопку «Отмена».

Форма добавления товара закрывается.

2.5.Поток ошибок.-

2.6.Постусловия.-

3. Редактирование товара.

3.1. Краткое описание. Администратор выбирает товар, который ему необходимо отредактировать и переходит на страницу редактирования товара, где можно изменить название товара, цену и количество.

3.2. Предусловия. Успешная аутентификация. Редактировать одновременно можно только 1 товар.

3.3. Основной и альтернативный потоки событий.

3.3.1. Администратор открывает форму просмотра каталога товаров.

3.3.2. Администратор «глазами» находит товар для редактирования и отмечает. Или альтернативный вариант 1.

3.3.3. Администратор нажимает кнопку «Редактировать» и получает форму редактирования товара, которая состоит из 3 полей: названия товара, цены и количества единиц. Все поля содержат имеющиеся значения.

3.3.4. Администратор исправляет значения и нажимает кнопку «Сохранить». Или альтернативный вариант 2.

3.3.5. Данные сохраняются в базу данных.

3.4.1 Альтернативный вариант событий 1.

Администратор использует функцию поиска, интегрированную со страницей просмотра каталога товаров. Вводит в поле для поиска название товара и нажимает кнопку «Искать».

Поисковый скрипт возвращает результат поиска вместо каталога товаров. Далее пункт 3.3.2. Или альтернативный вариант 3.

3.4.2. Альтернативный вариант событий 2. Администратор нажимает кнопку «Отмена» - форма редактирования закрывается.

3.4.3. Альтернативный вариант событий 3. Поиск не дал результатов. Администратор получает соответствующее сообщение и возвращается окно каталога товаров.

3.5.Поток ошибок.-

3.6.Постусловия.-

4. Удаление товара.

4.1. Краткое описание. Администратор выбирает товар, который ему необходимо удалить, а затем удаляет его.

4.2. Предусловия. Успешная аутентификация.

4.3. Основной поток событий.

4.3.1. Администратор открывает форму просмотра каталога товаров.

4.3.2. Администратор «глазами» находит товар для удаления и отмечает. Или альтернативный поток.

4.3.3. Администратор отмечает товары необходимые для удаления.

4.3.4.Администратор нажимает кнопку «Удалить». Товар(ы) удаляется из БД. Администратор возвращается в каталог товаров.

4.4.1. Альтернативный вариант событий 1.

Администратор использует функцию поиска, интегрированную со страницей просмотра каталога товаров. Вводит в поле для поиска название товара и нажимает кнопку «Искать».

Поисковый скрипт возвращает результат поиска вместо каталога товаров. Далее пункт 4.3.3. Или альтернативный вариант 2.

4.4.2. Альтернативный вариант событий 3. Поиск не дал результатов. Администратор получает соответствующее сообщение и возвращается окно каталога товаров.

4.5.Поток ошибок.-

4.6.Постусловия.-

5. Оформить продажу.

5.1. Краткое описание. Продавец оформляет продажу товара (исполнение заказа).

5.2. Предусловия. Успешная аутентификация. Журнал заказов содержит не исполненные заказы. Одновременно можно исполнять 1 заказ.

5.3. Основной поток событий.

5.3.1. Продавец открывает журнал заказов и отмечает заказ для исполнения.

5.3.2. Продавец нажимает кнопку «Оформить продажу».

5.3.3. Заказ исчезает из журнала заказов (заказ получает статус - «исполнен»). На экран выводится счет на оплату товара, указанного в заказе и количество товара в каталоге товаров уменьшается на количество товара, указанное в заказе.

5.3.4. Журнал заказов обновляется.

5.4. Альтернативный вариант событий 1.-

5.5.Поток ошибок.

5.6.Постусловия.

6.Удаление заказа.

6.1. Краткое описание. Продавец может удалить заказ из журнала, в случае если покупатель отказался покупать.

6.2. Предусловия. Успешная аутентификация. Журнал заказов содержит не исполненные заказы.

6.3. Основной поток событий.

6.3.1. Продавец открывает журнал заказов.

6.3.2. Продавец отмечает в журнале заказы, которые подлежат удалению.

6.3.3. Продавец нажимает кнопку «Удалить» и заказы удаляются из таблицы «Заказы».

6.4. Альтернативный вариант событий 1.-

6.5.Поток ошибок.-

6.6.Постусловия.-

7. Добавление товара в корзину

7.1. Краткое описание. Покупатель выбирает товар в каталоге товаров, указывает количество единиц и добавляет его в корзину.

7.2. Предусловия. -

7.3. Основной и альтернативный потоки событий.

7.3.1. Покупатель открывает каталог товаров. Сначала корзина пустая.

7.3.2. Покупатель «глазами» находит товар который хочет купить. Или альтернативный поток 1.

7.3.3. Покупатель отмечает товар для покупки и указывает количество товара, который хочет приобрести.

7.3.4. Покупатель нажимает кнопку - «Добавить в корзину» и товар добавляется в корзину.

7.3.5. Повторение пункта 7.3.2. или переход в просмотр корзины.

7.4.1. Альтернативный вариант событий 1.

Покупатель использует функцию поиска, интегрированную со страницей просмотра каталога товаров. Вводит в поле для поиска название товара и нажимает кнопку «Искать».

Поисковый скрипт возвращает результат поиска вместо каталога товаров. Далее пункт 7.3.3. Или альтернативный вариант 2.

7.4.2. Альтернативный вариант событий 2.

Альтернативный вариант событий 3. Поиск не дал результатов. Покупатель получает соответствующее сообщение и возвращается окно каталога товаров.

7.5.Поток ошибок.-

7.6.Постусловия.-

8.Удаление товара из корзины.

8.1. Краткое описание. Пользователь может отказаться от части покупок, для этого он должен удалить не понравившиеся товары из корзины.

8.2. Предусловия. Не пустая корзина.

8.3. Основной поток событий.

8.3.1.Покупатель использует функцию просмотра корзины, интегрированную со страницей просмотра каталога товаров. Покупатель нажимает кнопку «Открыть корзину».

8.3.2. Открывается окно с содержимым корзины. На окне имеются 3 кнопки «Удалить», «Оформить заказ» и «Закрыть корзину».

8.3.3. Покупатель отмечает товары, которые ему не понравились и нажимает кнопку «Удалить». Или альтернативный вариант событий.

8.3.4. Окно корзины перезагружается и будет содержать только нужные покупателю товары.

8.4.1. Альтернативный вариант событий 1.

Покупатель нажимает кнопку «Закрыть корзину», при этом окно корзины закрывается и на экране появляется каталог товаров.

8.5.Поток ошибок.

8.6.Постусловия.

9. Оформление заказа.

9.1. Краткое описание. Покупатель отредактировал корзину и готов произвести заказ. Заказ будет принят магазином только после регистрации покупателя.

9.2. Предусловия. Корзина не пустая.

9.3. Основной и альтернативный потоки событий.

9.3.1. Покупатель открывает корзину из каталога товаров

9.3.2. Покупатель нажимает кнопку «Оформить заказ».

9.3.3. На экране появляется форма регистрации покупателя, которая содержит поля для ввода ФИО, телефона и адреса доставки и кнопку «Зарегистрироваться».

9.3.4. Пользователь вводит данные и нажимает кнопку «Зарегистрироваться». Программа проверяет наличие этого покупателя в таблице покупателей и если такого нет, то вводит его вновь и запоминает идентификатор, если он есть, то запоминает его идентификатор.

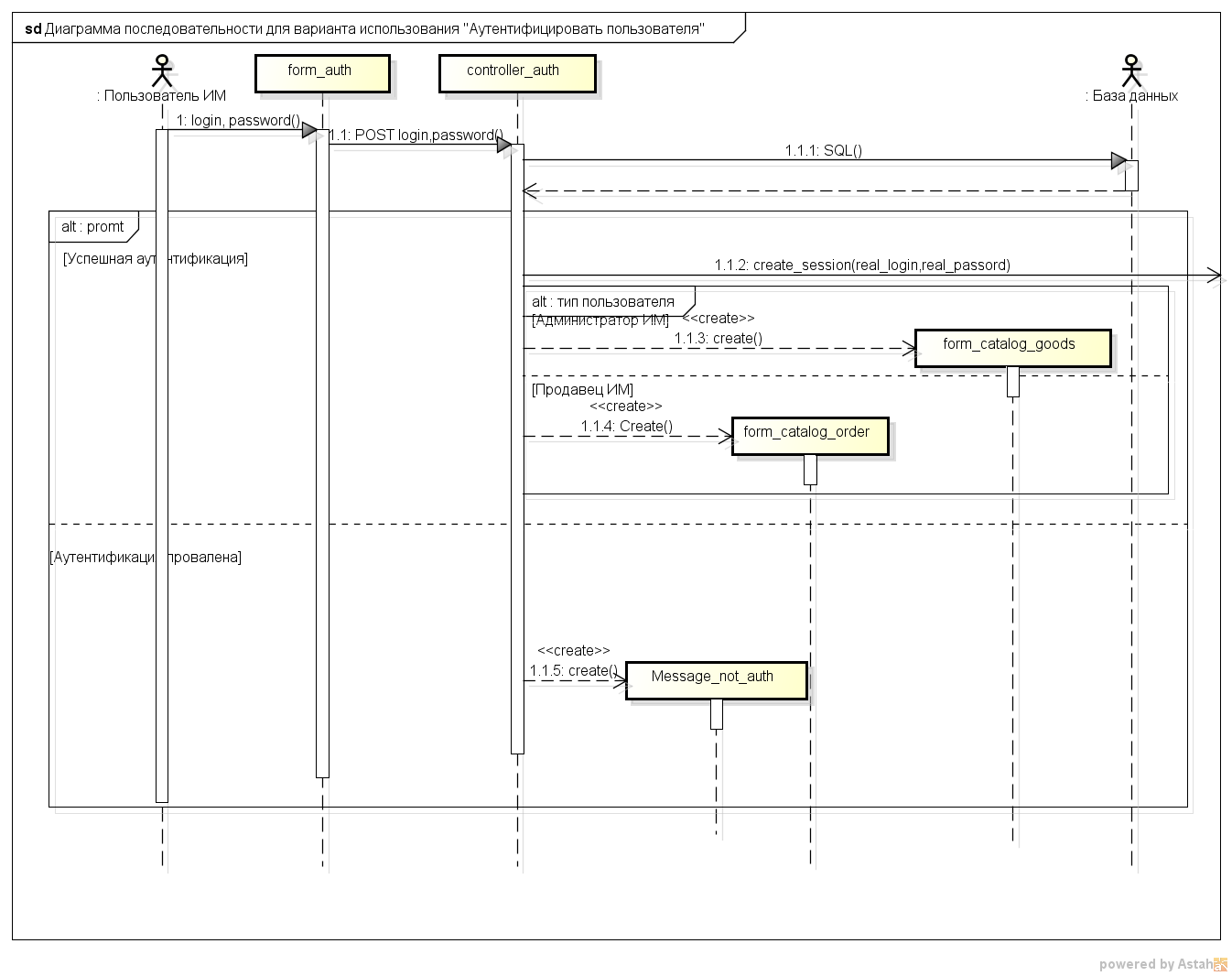
9.3.5. После регистрации программа сохраняет товары из корзины в таблицу заказов и выводит на экран сообщение, что заказ принят и показывает номер заказа.

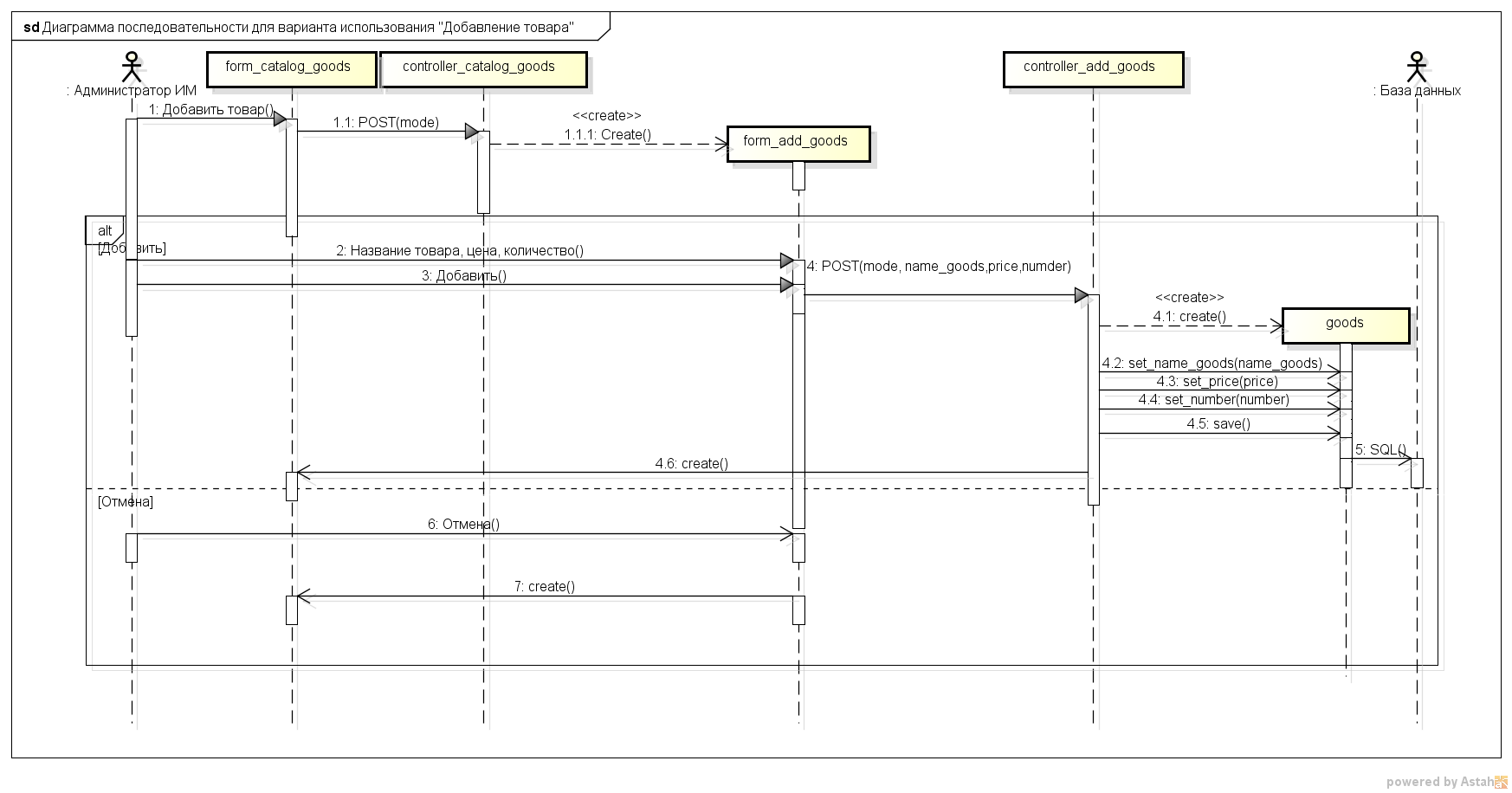
9.4. Альтернативный вариант событий 1.-

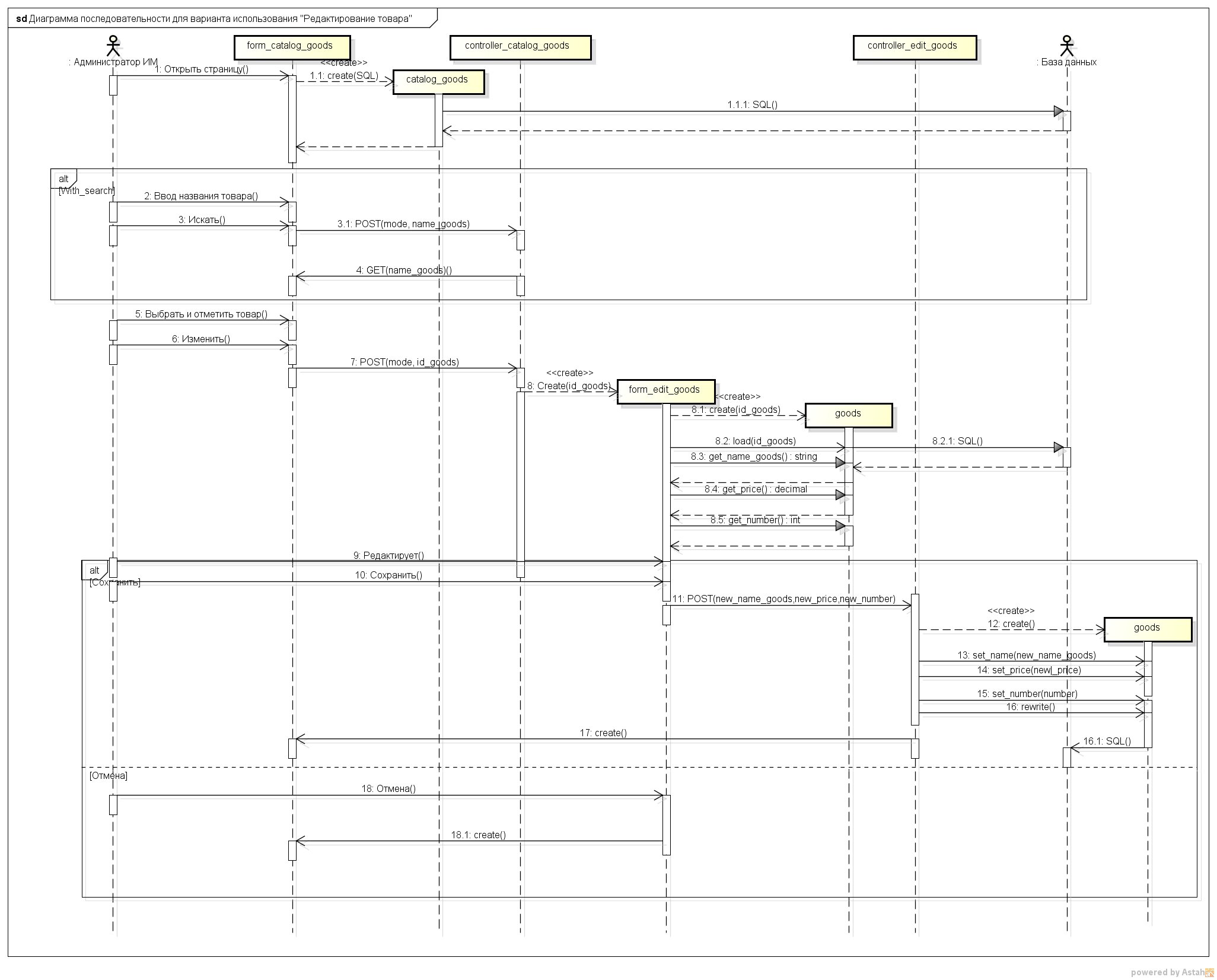
9.5.Поток ошибок.

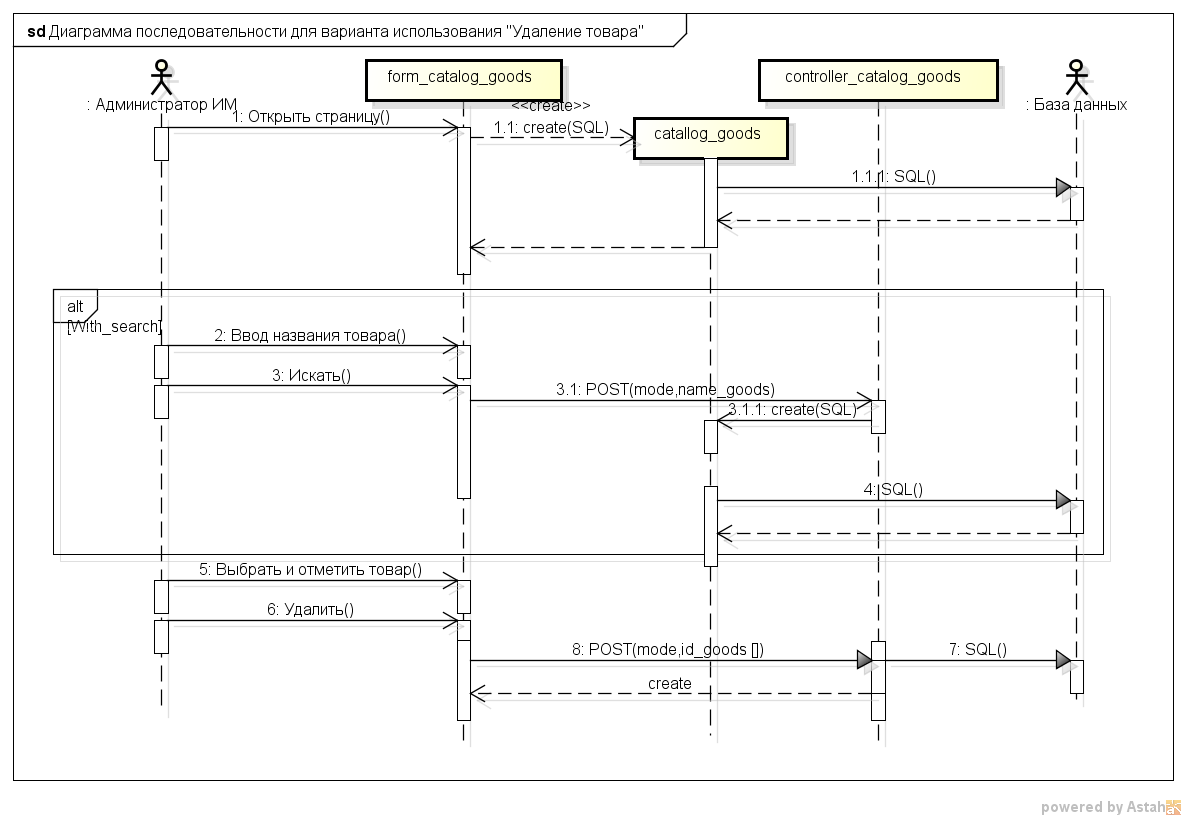
9.6.Постусловия.

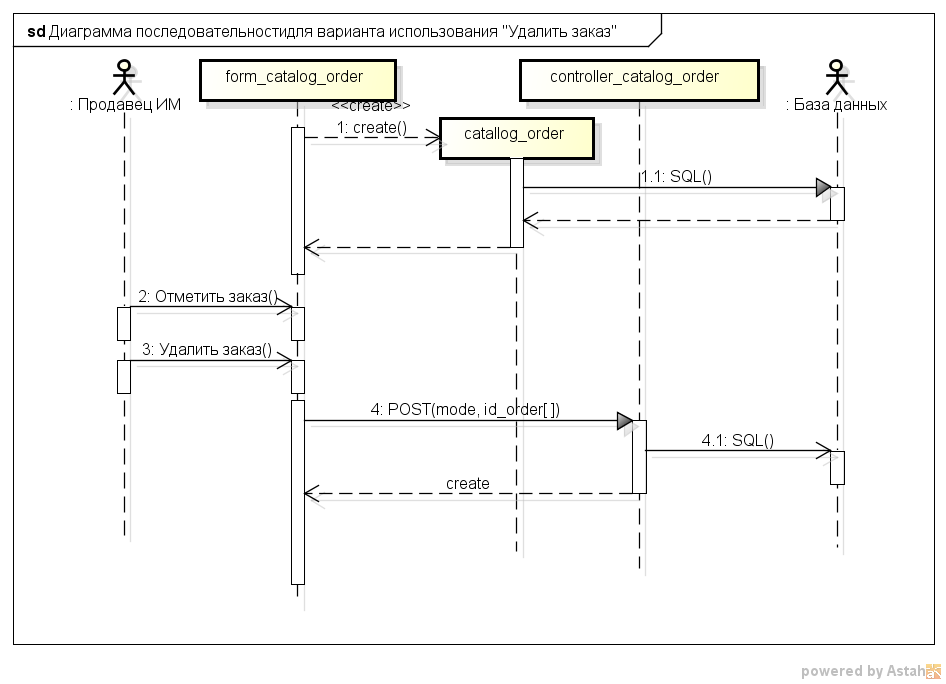
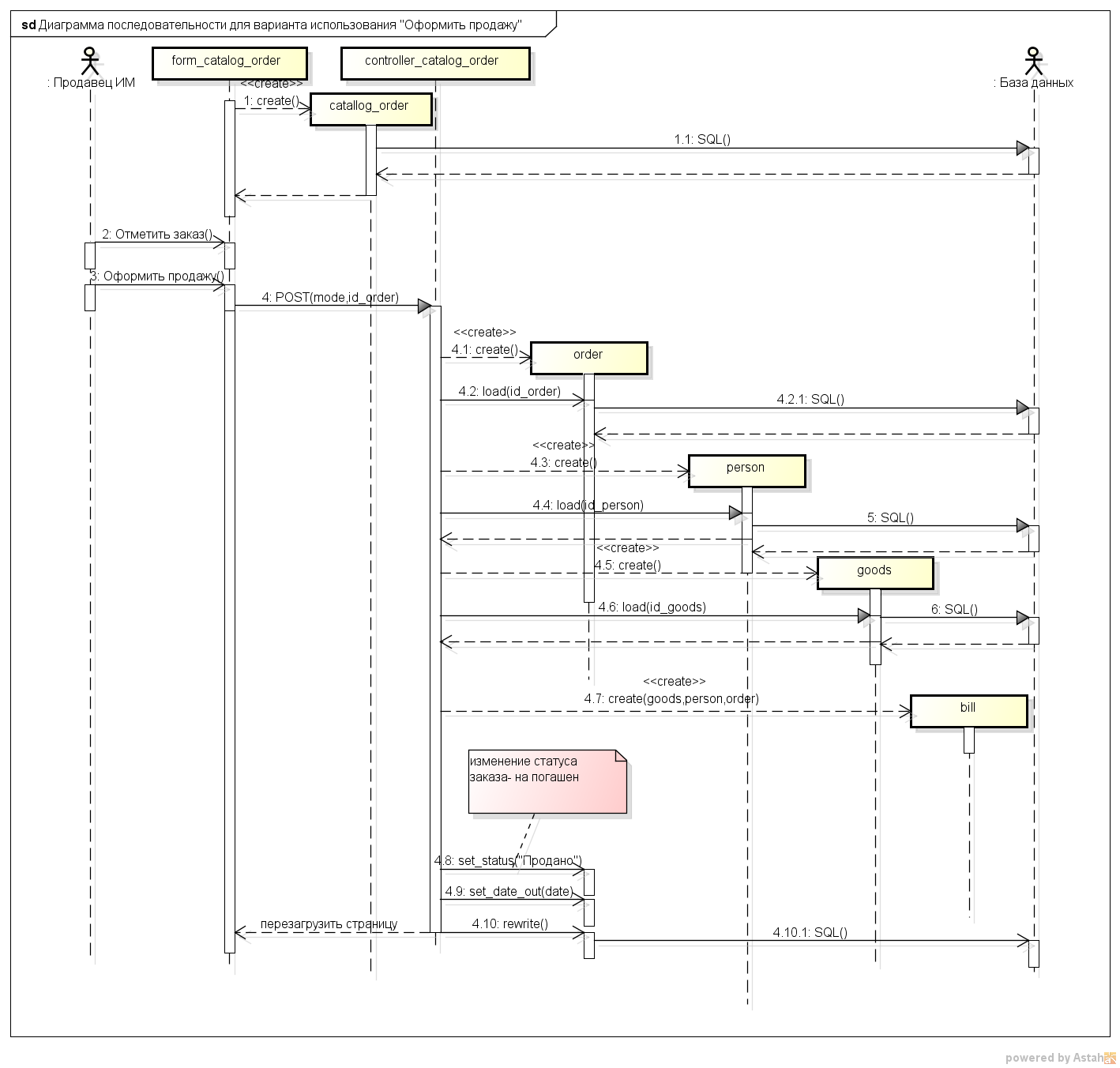
Для выделения объектов и их взаимодействия создадим диаграммы последовательности для каждого рассмотренного потока событий.

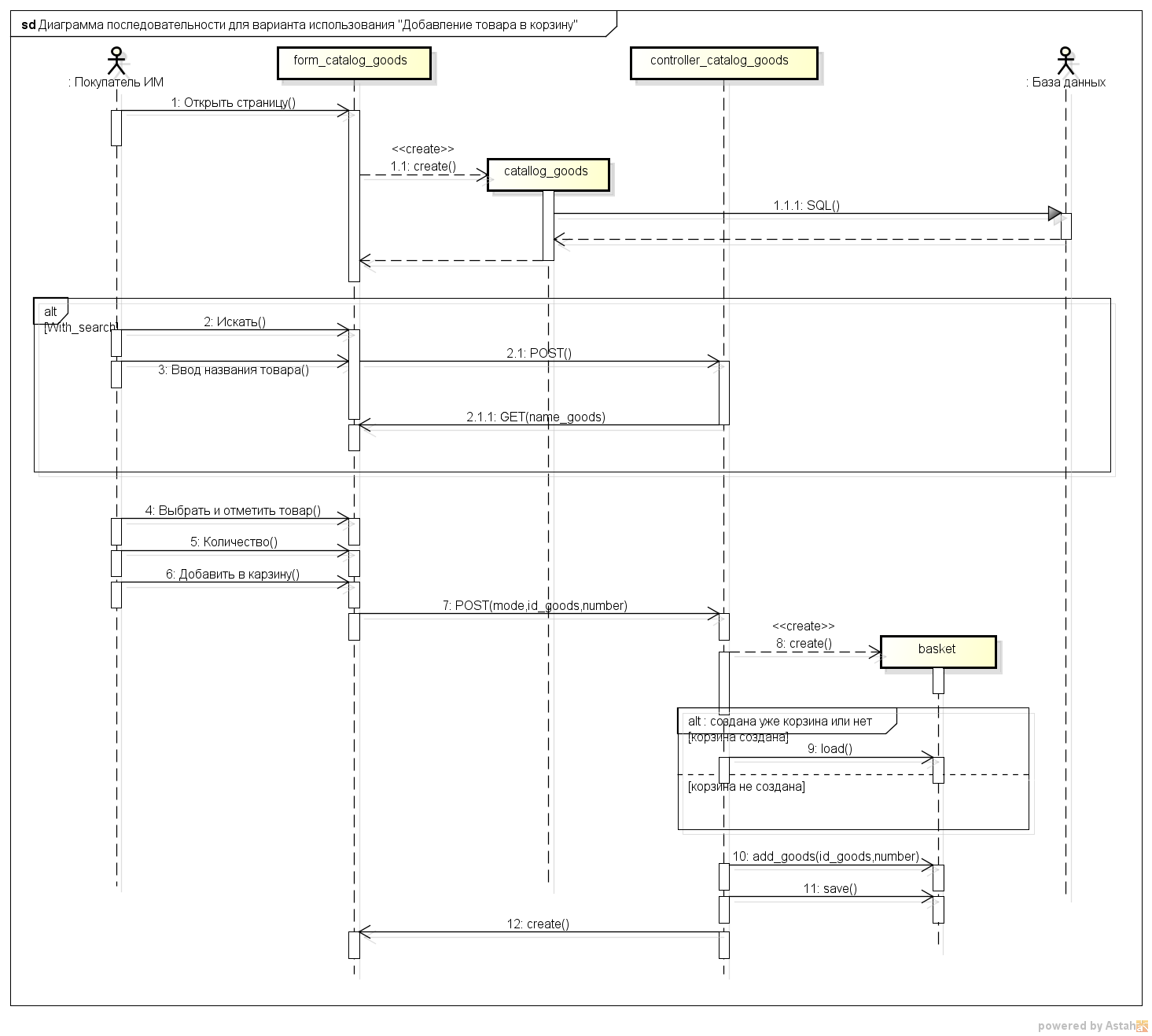


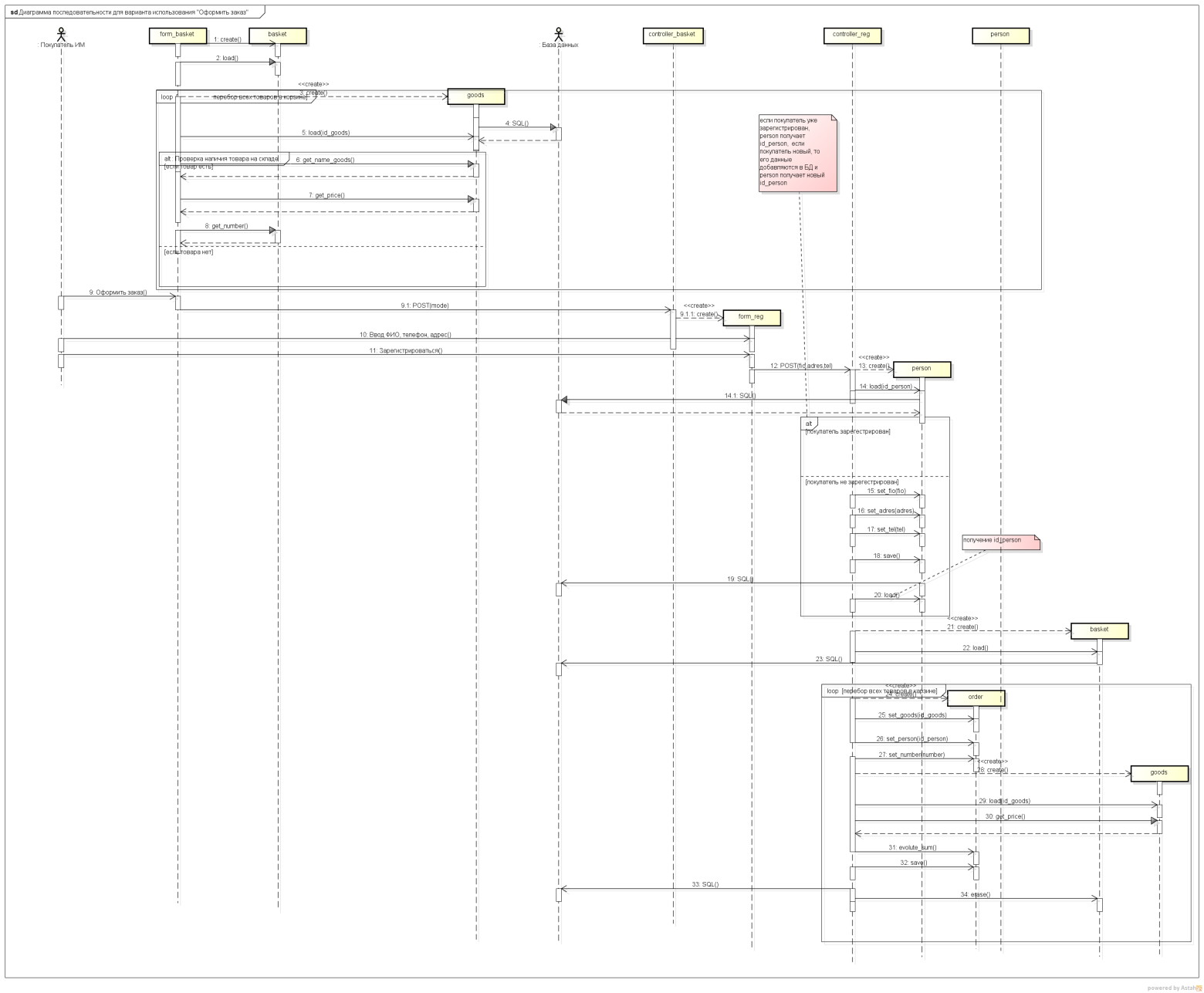


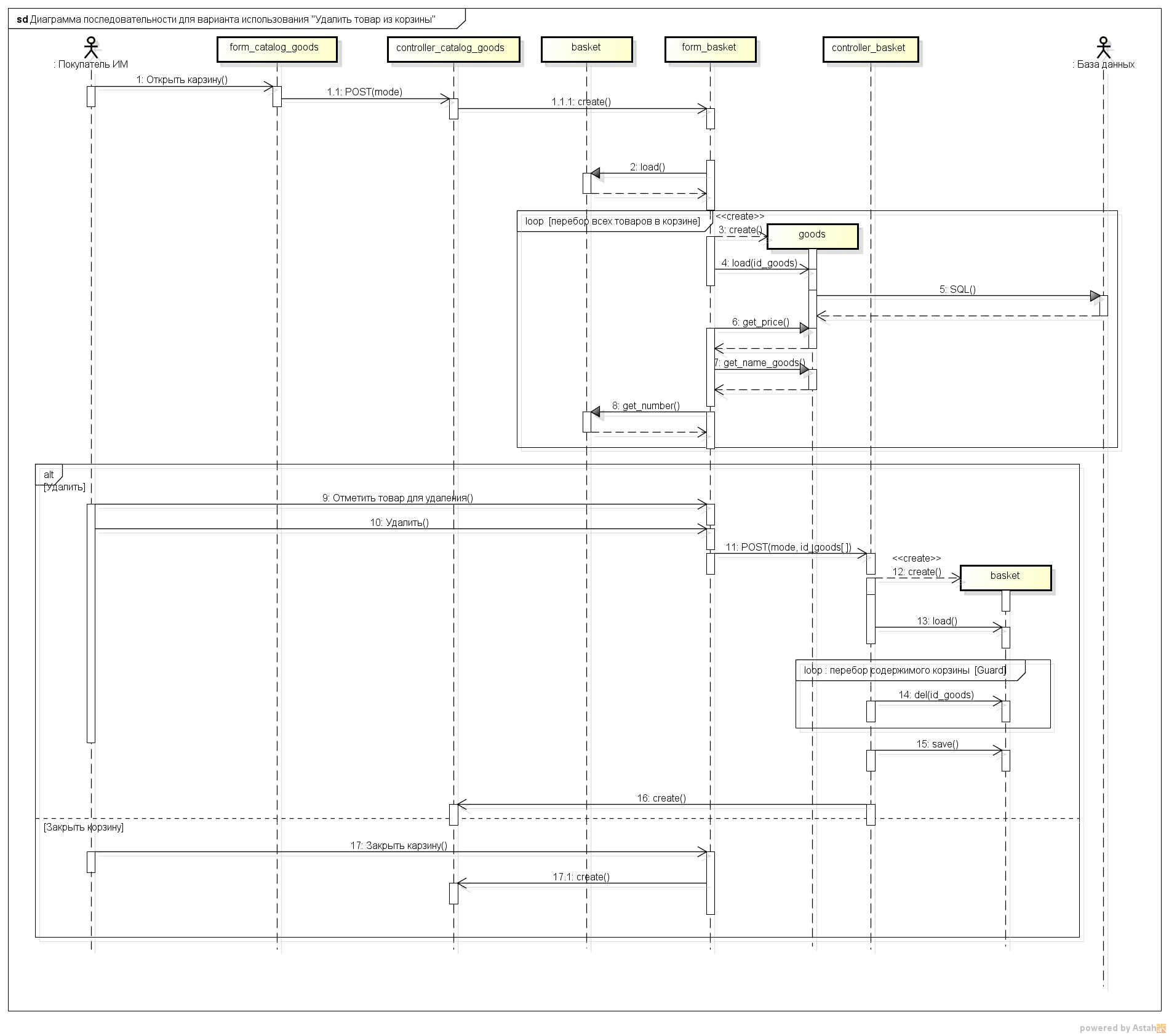








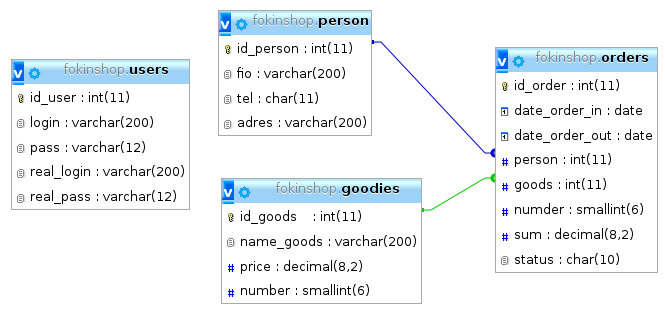




## 2.4. Разработка структуры хранения данных проекта

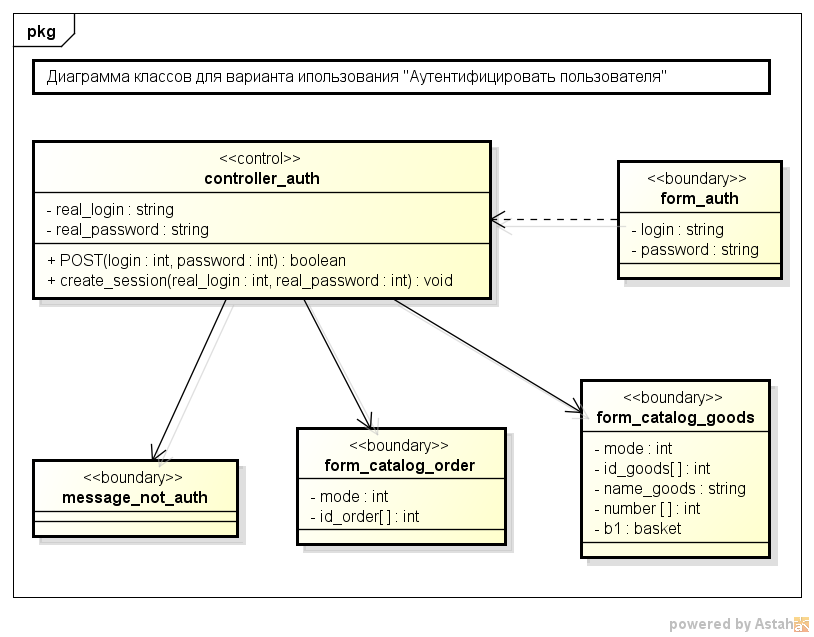
Хранение данных проекта предполагается в базе данных сервера mysql 5.

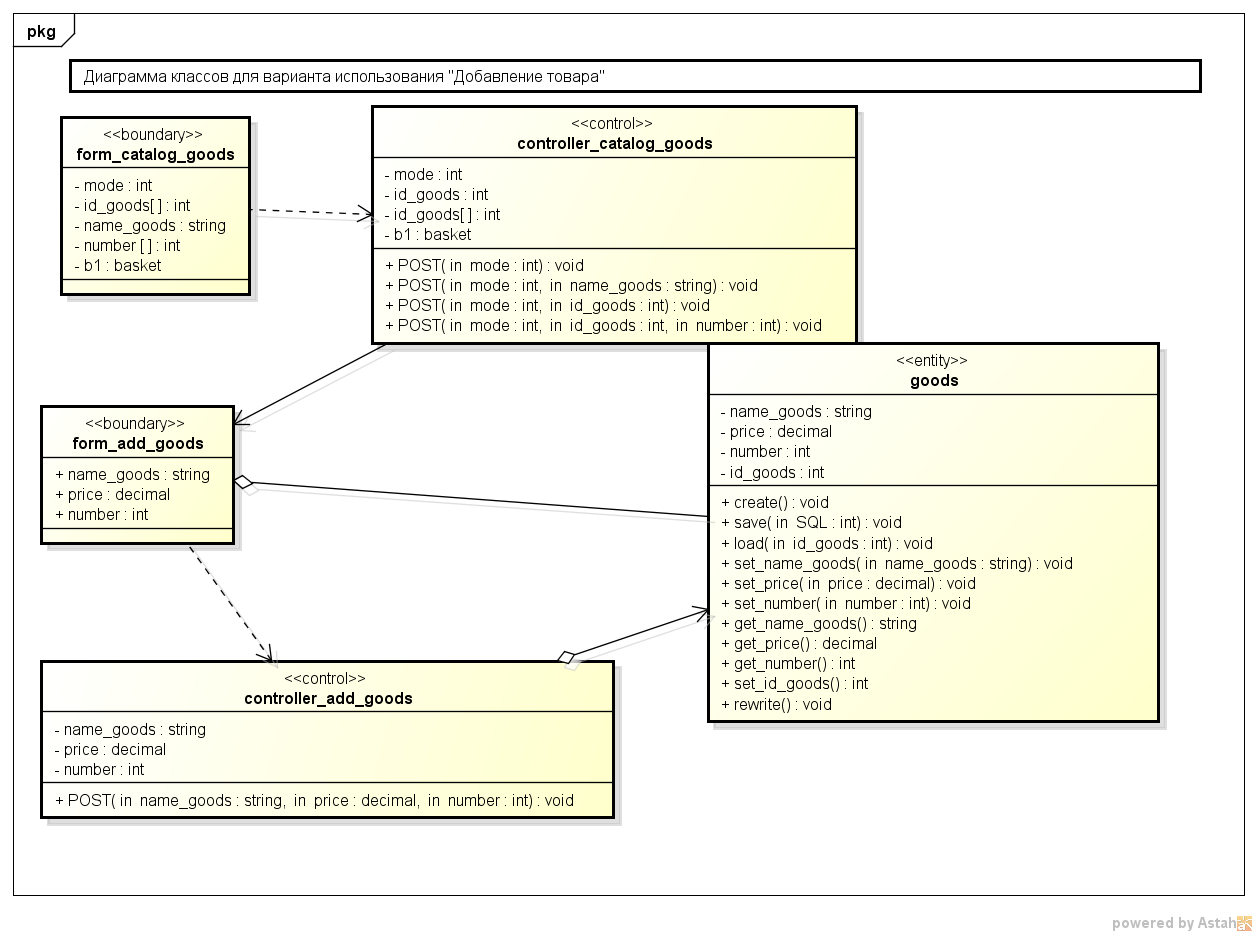
В базе данных предполагается хранение таких сущностей как: Пользователи, Покупатели, Товары, Заказы. Для каждой сущности существует своя таблица: Пользователи- users, Покупатели - person, Товары- goodies, Заказы- orders:

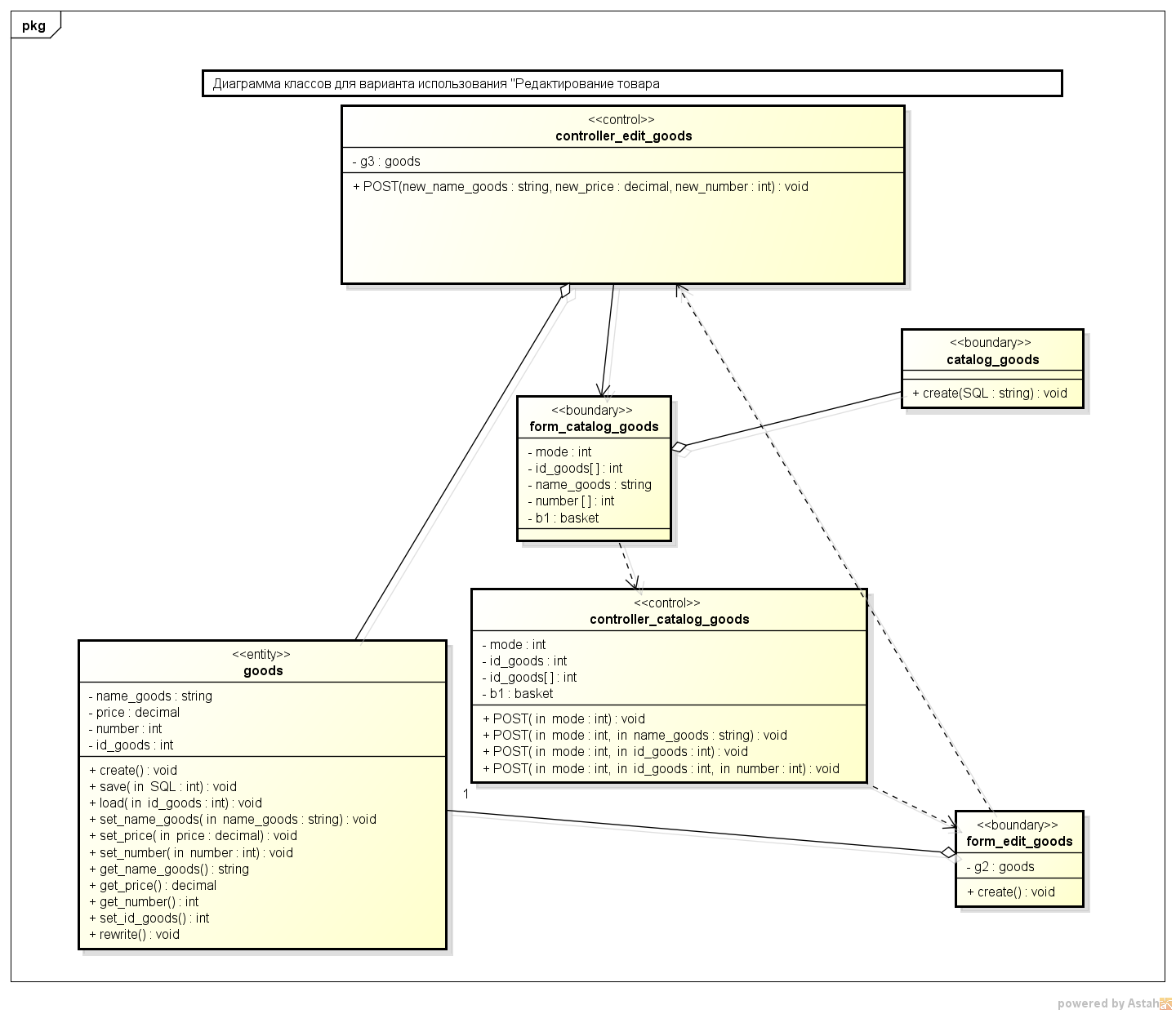


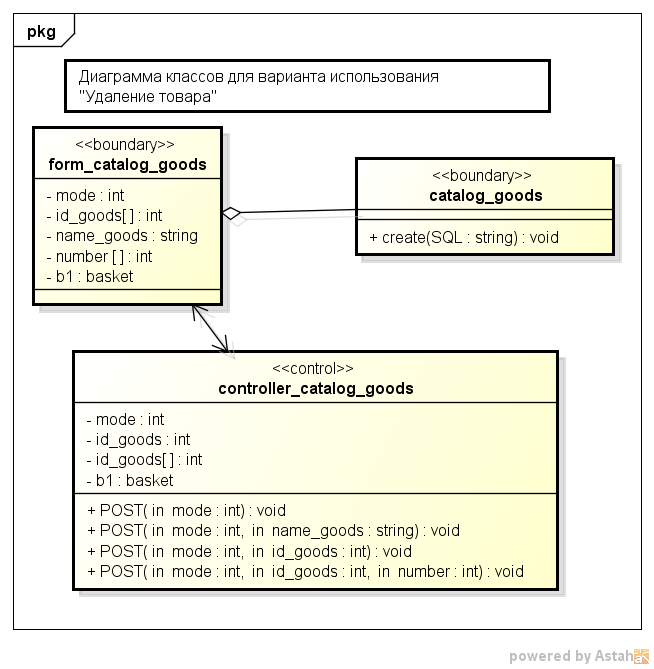
Cookies “basket” содержит информацию о содержании корзины пользователя, является 2-мерным массивом. Массив содержит информацию о товарах, которые выбрал покупатель, которая тоже является массивом и имеет 2 поля соответствующие id\_goods – индексу выбранного товара и number – количеству заказанного товара.

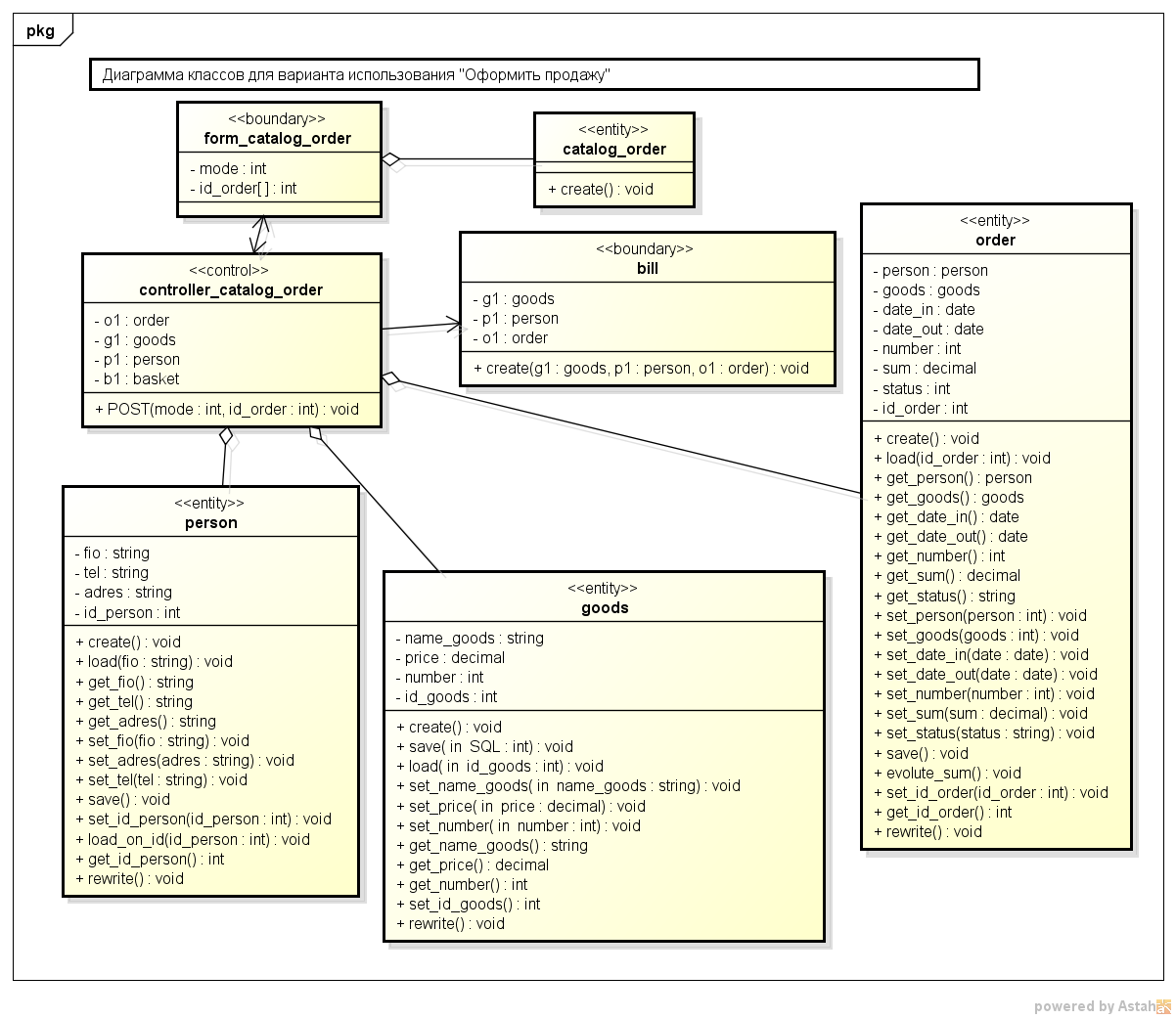
## 2.5. Разработка структуры классов для обработки данных.

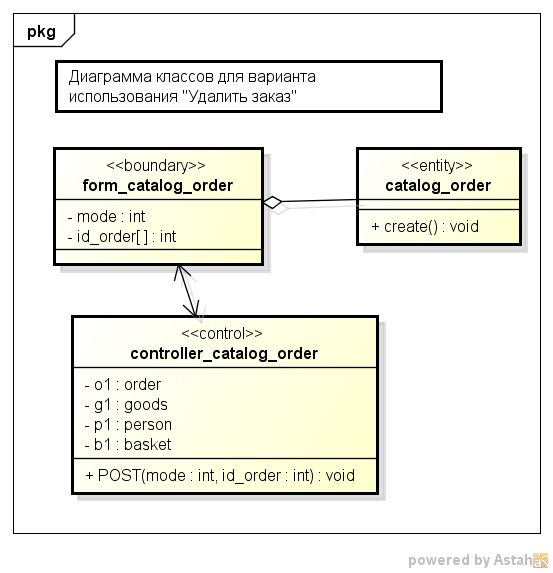


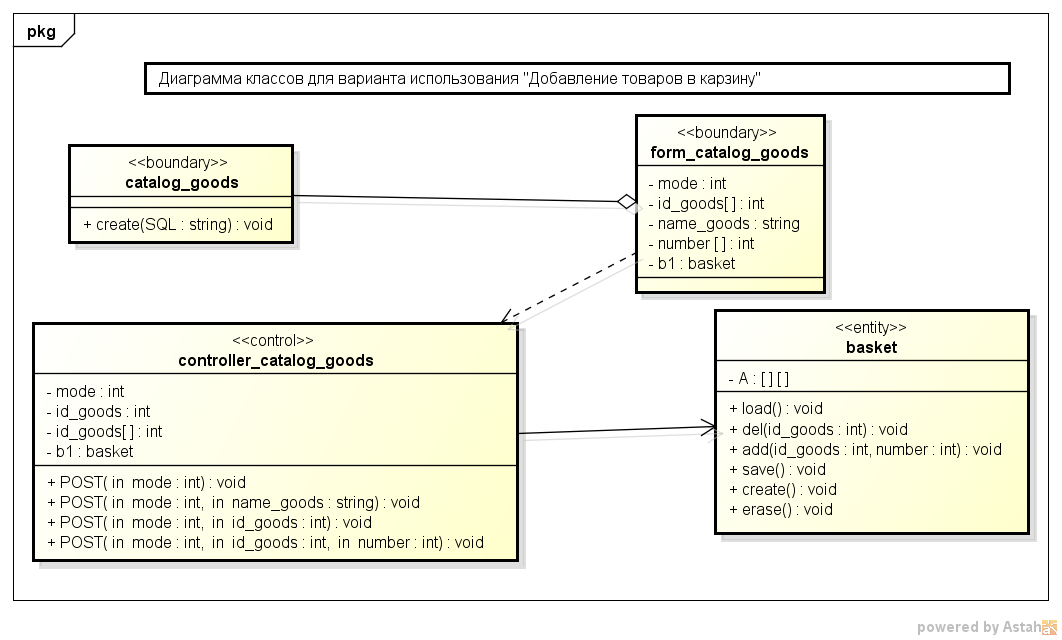


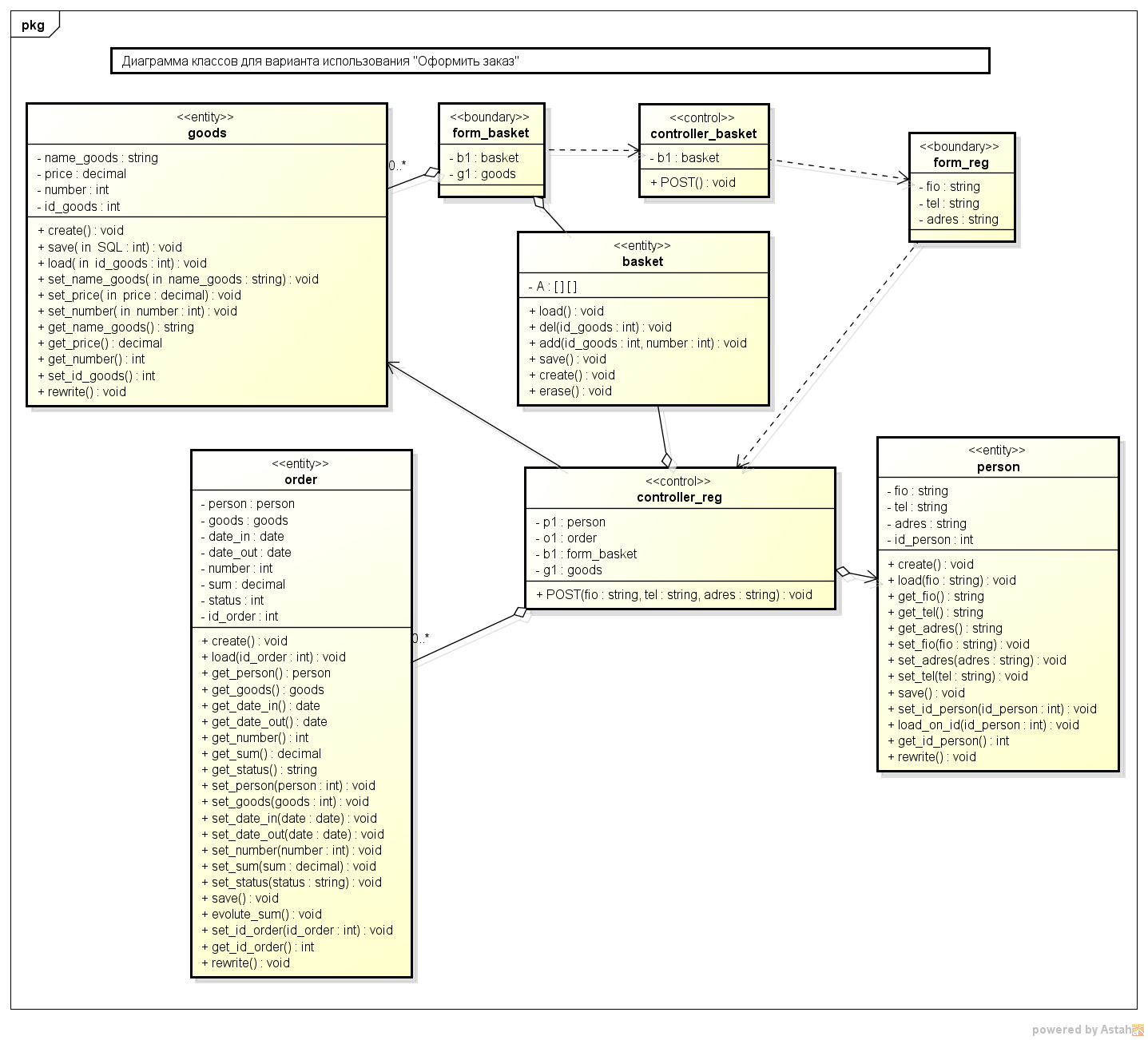


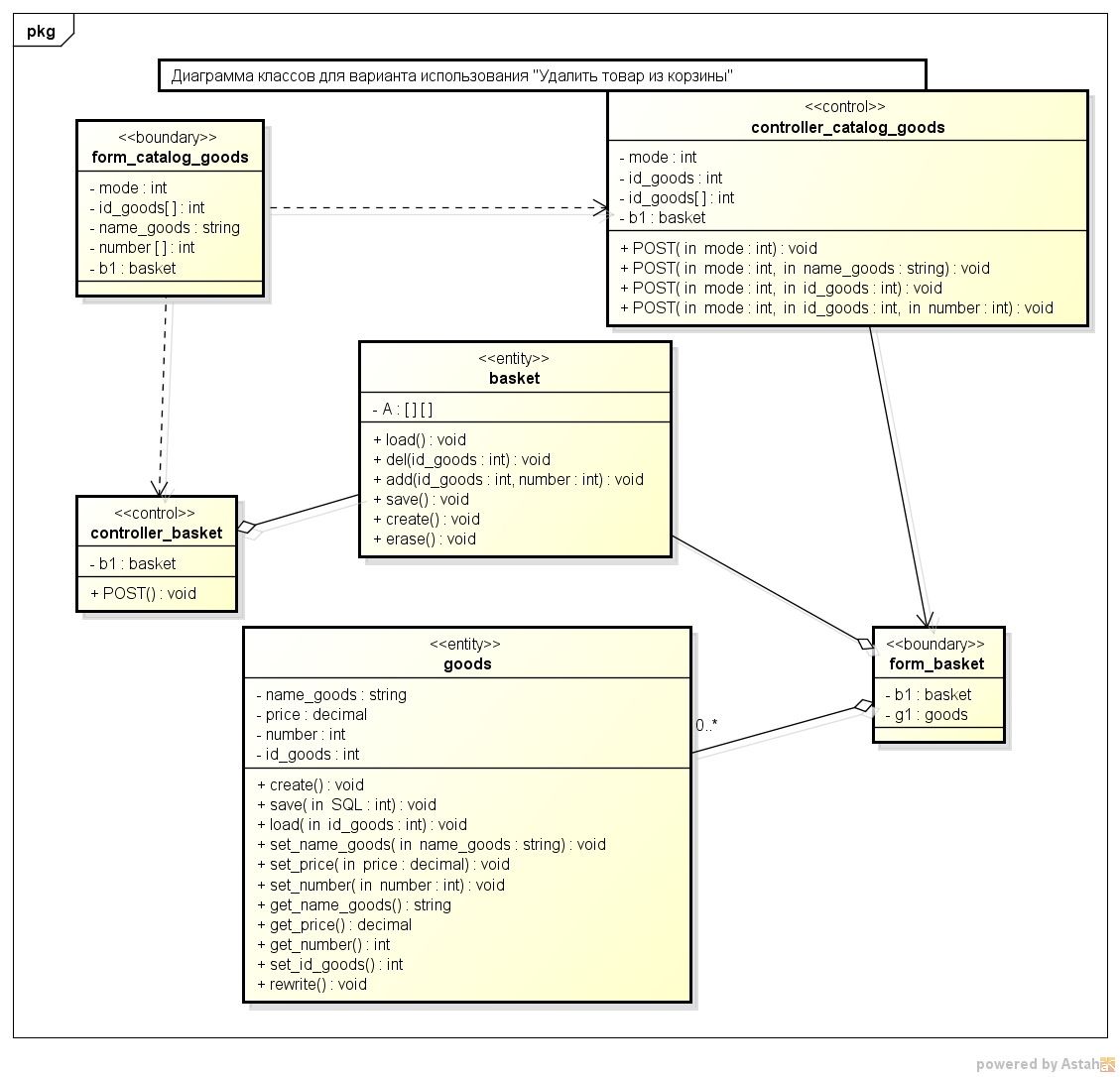












## 2.6. Формирование перечня интерфейсов для ввода данных и определение заполняемых полей в каждом интерфейсе

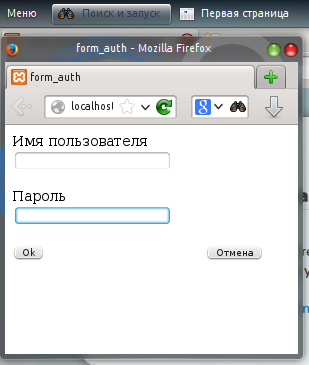
Сведем все граничные классы и их свойства в одну таблицу:

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п. | Название класса | Описание | Заполняемые поля |
| 1. | form\_auth | Форма для ввода имени пользователя и пароля для аутентификации пользователя интернет магазина | login-Имя пользователя  password - пароль |
| 2. | form\_catalog\_goods | Форма отображения товаров в интернет магазине. В зависимости от пользователя имеет разный вид. Покупатель и Продавец могут только выбирать товар используя radiobutton и указывать его количество в единицах. Администратор может помечать товар для удаления, используя cheсkbox компоненты, выбирать товар для редактирования, используя radiobutton. Для всех пользователей предоставляется возможность поиска товара по названию. | id\_goods – индекс выбранного Покупателем товара или товара, выбранного Администратором для редактирования.  id\_goods [] - массив индексов товаров, выбранных Администратором для удаления;  number – количество товара, выбранного покупателем;  name\_goods – название товара для поиска. |
| 3. | message\_not\_auth | Форма для отображения сообщения о том, что неверно введены имя пользователя и пароль. | - |
| 4. | form\_basket | Форма для отображения состояния корзины. Покупатель может помечать товар для удаления из корзины, используя cheсkbox компоненты. | id\_goods [] - массив индексов товаров, выбранных Покупателем для удаления из корзины или для оформления заказа; |
| 5. | form\_reg | Форма регистрации Покупателя. Содержит поля fio, tel, adres. | Fio- фамилия, имя и отчество Покупателя, tel — телефон Покупателя,  adres - адрес Покупателя. |
| 6. | form\_edit\_goods | Форма редактирования товара. Используется администратором для изменения атрибутов товара (названия, цены и количества). | new\_name\_goods – новое название товара;  new\_price – новая цена товара;  new\_number – новое количество товара. |
| 7. | form\_catalog\_order | Форма просмотра заказов используется Продавцом для удаления заказов и оформления продажи. Продавец может помечать заказы для удаления, используя cheсkbox компоненты и выбирать заказ для оформления продажи, используя radiobutton. | id\_order – индекс заказа для оформления продажи;  id\_order [] - массив индексов заказов для их удаления. |
| 8. | form\_add\_goods | Форма для ввода нового товара в каталог. Используется Администратором. Администратор вводит в соответствующие поля данные по новому товару и нажимает кнопку «Добавить». | new\_name\_goods — название нового товара;  new\_price — цена нового товара;  new\_number — количество нового товара. |

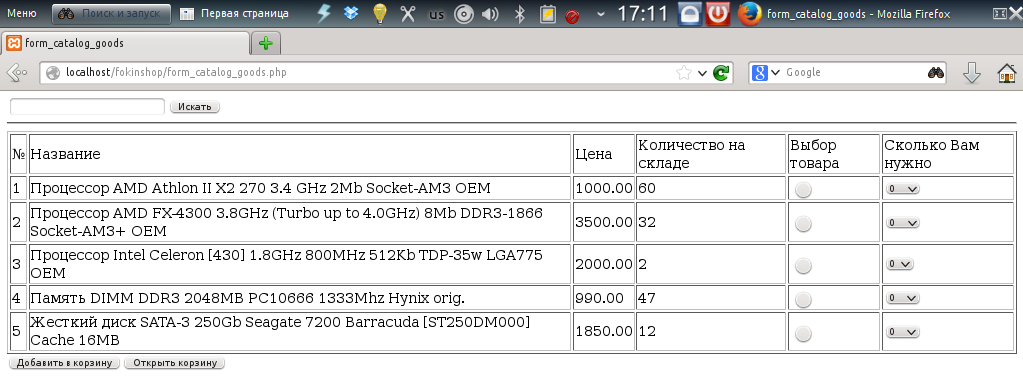
## 2.7. Создание статических страниц для каждого раздела

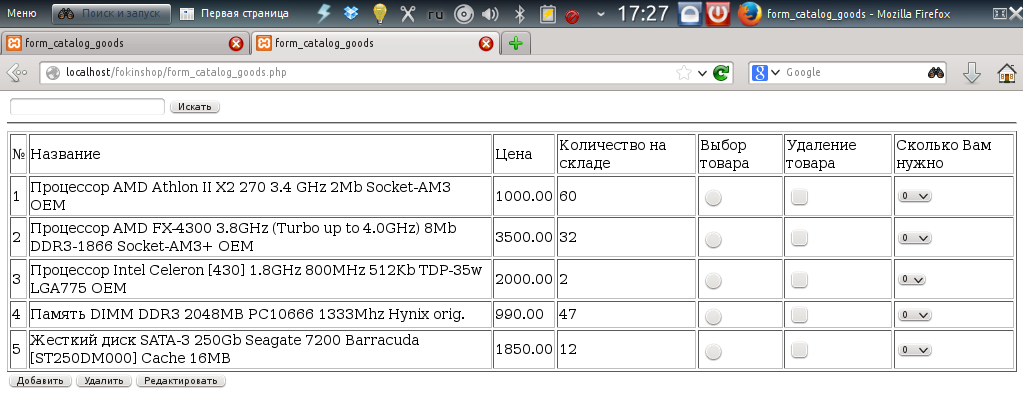
Для создания использовалось средство разработки BlueGriffon.

1. form\_auth

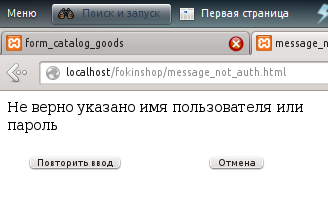


2. form\_catalog\_goods

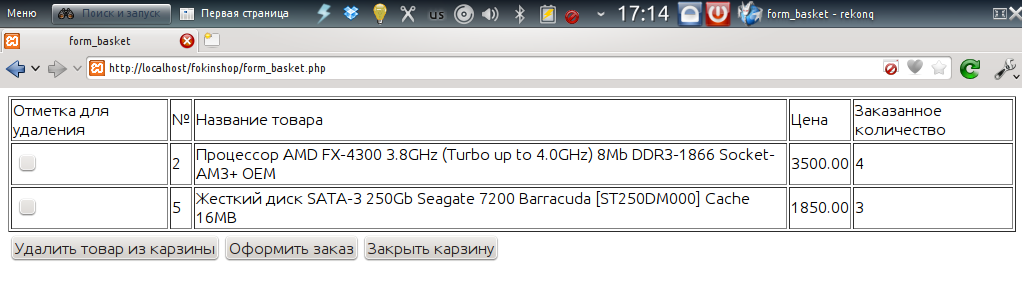




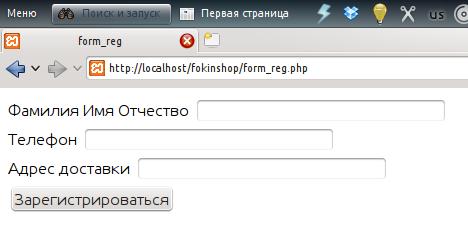
3. message\_not\_auth

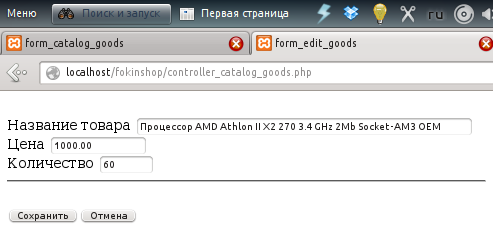


4. form\_basket

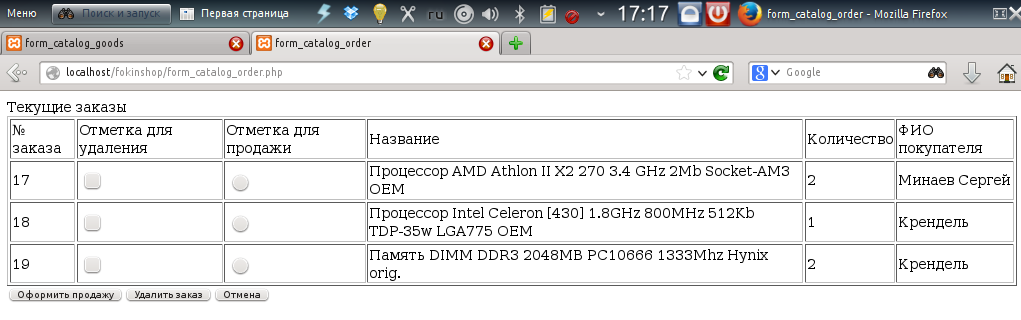


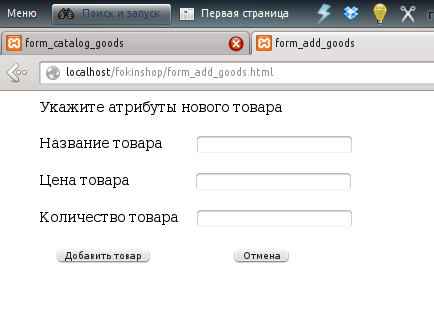
5. form\_reg

  
  
6. form\_edit\_goods

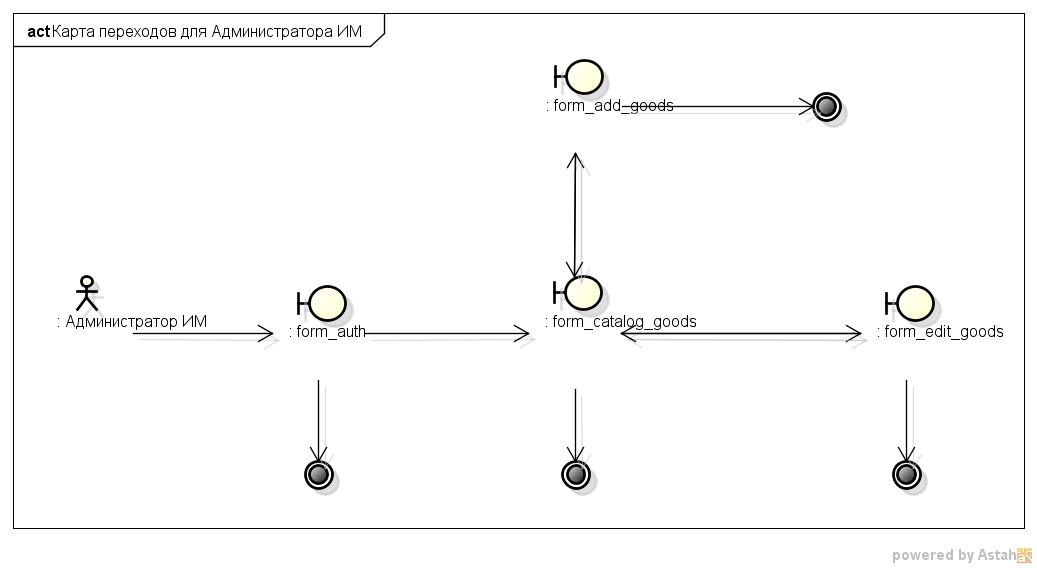


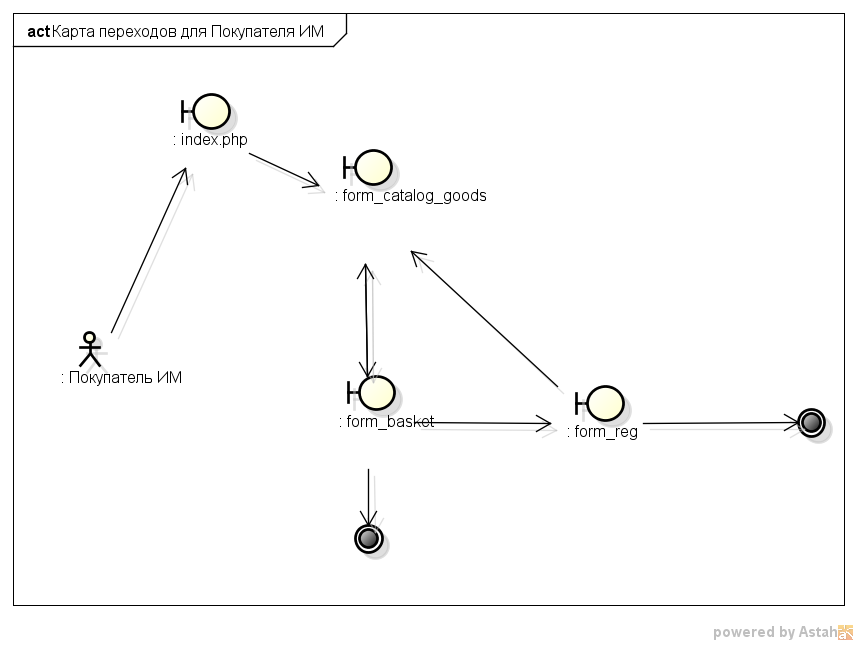
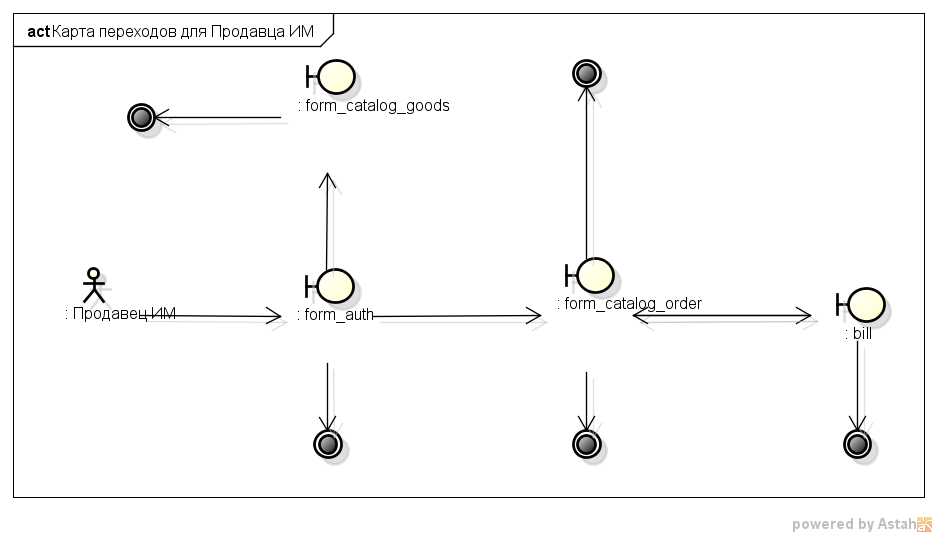
7. form\_catalog\_order

  
8. form\_add\_goods



## 2.8. Проектирование карты переходов





# 3. Создание базы данных

Создание базы данных выполнено в редактор phpmyAdmin.

Итоговый скрипт базы данных с тестовыми записями и 4 пользователями:

-- phpMyAdmin SQL Dump

-- version 4.0.4.1

-- http://www.phpmyadmin.net

--

-- Хост: localhost

-- Время создания: Авг 24 2013 г., 09:27

-- Версия сервера: 5.5.31

-- Версия PHP: 5.4.16

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

--

-- База данных: `fokinshop`

--

CREATE DATABASE IF NOT EXISTS `fokinshop` DEFAULT CHARACTER SET cp1251 COLLATE cp1251\_general\_cs;

USE `fokinshop`;

-- --------------------------------------------------------

--

-- Структура таблицы `goodies`

--

CREATE TABLE IF NOT EXISTS `goodies` (

`id\_goods` int(11) NOT NULL AUTO\_INCREMENT,

`name\_goods` varchar(250) COLLATE cp1251\_general\_cs NOT NULL,

`price` decimal(8,2) NOT NULL,

`number` smallint(6) NOT NULL,

PRIMARY KEY (`id\_goods`)

) ENGINE=MyISAM DEFAULT CHARSET=cp1251 COLLATE=cp1251\_general\_cs AUTO\_INCREMENT=14 ;

--

-- Дамп данных таблицы `goodies`

--

INSERT INTO `goodies` (`id\_goods`, `name\_goods`, `price`, `number`) VALUES

(1, 'Процессор AMD Athlon II X2 270 3.4 GHz 2Mb Socket-AM3 OEM', '1000.00', 60),

(2, 'Процессор AMD FX-4300 3.8GHz (Turbo up to 4.0GHz) 8Mb DDR3-1866 Socket-AM3+ OEM', '3500.00', 32),

(3, 'Процессор Intel Celeron [430] 1.8GHz 800MHz 512Kb TDP-35w LGA775 OEM', '2000.00', 2),

(4, 'Память DIMM DDR3 2048MB PC10666 1333Mhz Hynix orig.', '990.00', 47),

(5, 'Жесткий диск SATA-3 250Gb Seagate 7200 Barracuda [ST250DM000] Cache 16MB', '1850.00', 12);

-- --------------------------------------------------------

--

-- Структура таблицы `order`

--

CREATE TABLE IF NOT EXISTS `order` (

`id\_order` int(11) NOT NULL AUTO\_INCREMENT,

`date\_in` date NOT NULL,

`date\_out` date NOT NULL,

`person` int(11) NOT NULL,

`goods` int(11) NOT NULL,

`sum` decimal(10,0) NOT NULL,

`number` int(11) NOT NULL,

`status` varchar(200) COLLATE cp1251\_general\_cs NOT NULL,

PRIMARY KEY (`id\_order`)

) ENGINE=MyISAM DEFAULT CHARSET=cp1251 COLLATE=cp1251\_general\_cs AUTO\_INCREMENT=17 ;

--

-- Дамп данных таблицы `order`

--

INSERT INTO `order` (`id\_order`, `date\_in`, `date\_out`, `person`, `goods`, `sum`, `number`, `status`) VALUES

(14, '2013-08-24', '2013-08-24', 16, 3, '4000', 2, 'Продано'),

(13, '2013-08-24', '2013-08-24', 15, 1, '2000', 2, 'Продано'),

(11, '2013-08-24', '2013-08-24', 14, 5, '3700', 2, 'Продано'),

(10, '2013-08-24', '2013-08-24', 14, 2, '14000', 2, 'Продано'),

(9, '2013-08-24', '2013-08-24', 13, 3, '2000', 1, 'Продано'),

(15, '2013-08-24', '2013-08-24', 16, 4, '4950', 5, 'Продано'),

(16, '2013-08-24', '2013-08-24', 16, 1, '2000', 2, 'Продано');

-- --------------------------------------------------------

--

-- Структура таблицы `person`

--

CREATE TABLE IF NOT EXISTS `person` (

`id\_person` int(11) NOT NULL AUTO\_INCREMENT,

`fio` varchar(200) COLLATE cp1251\_general\_cs NOT NULL,

`tel` char(11) COLLATE cp1251\_general\_cs NOT NULL,

`adres` varchar(200) COLLATE cp1251\_general\_cs NOT NULL,

PRIMARY KEY (`id\_person`)

) ENGINE=MyISAM DEFAULT CHARSET=cp1251 COLLATE=cp1251\_general\_cs AUTO\_INCREMENT=17 ;

--

-- Дамп данных таблицы `person`

--

INSERT INTO `person` (`id\_person`, `fio`, `tel`, `adres`) VALUES

(2, 'Фокин', '123456', 'Киров'),

(3, 'фокин2', '123456', 'Киров'),

(4, 'Fokin', '123456', 'Kirov'),

(5, 'Gtnz', '123456', 'Rjjr'),

(6, 'wwwwww', '2345', 'rtrt'),

(7, 'ert', '123', 'qwe'),

(8, 'ввввв', '344444', 'ккккк'),

(9, 'eeeeee', '34567', '1234'),

(10, '1111111', '1111111', '11111111'),

(11, 'Зеельман', '1234567', 'Рига'),

(12, 'Петри', '654321', 'Рига'),

(13, 'Ермаков Николай', '541097', 'Киров Некрасова 8-53'),

(14, 'Юлиан', '', ''),

(15, 'Смоленский', 'уапывамыв', 'ывмывмы'),

(16, 'Малевич', '123321', 'Москва');

-- --------------------------------------------------------

--

-- Структура таблицы `users`

--

CREATE TABLE IF NOT EXISTS `users` (

`id\_user` int(11) NOT NULL AUTO\_INCREMENT,

`login` varchar(200) COLLATE cp1251\_general\_cs NOT NULL,

`pass` varchar(200) COLLATE cp1251\_general\_cs NOT NULL,

`reallogin` varchar(200) COLLATE cp1251\_general\_cs NOT NULL,

`realpass` varchar(200) COLLATE cp1251\_general\_cs NOT NULL,

PRIMARY KEY (`id\_user`)

) ENGINE=MyISAM DEFAULT CHARSET=cp1251 COLLATE=cp1251\_general\_cs AUTO\_INCREMENT=3 ;

--

-- Дамп данных таблицы `users`

--

INSERT INTO `users` (`id\_user`, `login`, `pass`, `reallogin`, `realpass`) VALUES

(1, 'Иванов', '123', 'fokinshopadmin', '123'),

(2, 'Петров', '123', 'fokinshopseller', '123');

-- пользователи

--администратор

GRANT USAGE ON \*.\* TO 'fokinshopadmin'@'localhost' IDENTIFIED BY PASSWORD '\*23AE809DDACAF96AF0FD78ED04B6A265E05AA257';

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON `fokinshop`.`goodies` TO 'fokinshopadmin'@'localhost';

--продавец

GRANT USAGE ON \*.\* TO 'fokinshopseller'@'localhost' IDENTIFIED BY PASSWORD '\*23AE809DDACAF96AF0FD78ED04B6A265E05AA257';

GRANT SELECT, UPDATE, DELETE ON `fokinshop`.`order` TO 'fokinshopseller'@'localhost';

GRANT SELECT, UPDATE ON `fokinshop`.`goodies` TO 'fokinshopseller'@'localhost';

GRANT SELECT ON `fokinshop`.`person` TO 'fokinshopseller'@'localhost';

--для авторизации

GRANT USAGE ON \*.\* TO 'fokinshoptemp'@'localhost' IDENTIFIED BY PASSWORD '\*23AE809DDACAF96AF0FD78ED04B6A265E05AA257';

GRANT SELECT ON `fokinshop`.`users` TO 'fokinshoptemp'@'localhost';

--покупатель

GRANT USAGE ON \*.\* TO 'fokinshopuser'@'localhost' IDENTIFIED BY PASSWORD '\*23AE809DDACAF96AF0FD78ED04B6A265E05AA257';

GRANT SELECT ON `fokinshop`.`goodies` TO 'fokinshopuser'@'localhost';

GRANT SELECT, INSERT ON `fokinshop`.`person` TO 'fokinshopuser'@'localhost';

GRANT INSERT ON `fokinshop`.`order` TO 'fokinshopuser'@'localhost';

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

В таблице users введено 2 пользователя:

Иванов с паролем «123» - администратор магазина и Петров с паролем «123» - продавец.

# 4. Разработка

Скрипты и дамп базы данных приведены во вложении.

Скрипты написаны на PHP 5.4.16. Нет полной совместимости с предыдущими версиями PHP.

# 5. Заключение

1. Результатом работ над курсовым проектом являются набор php - скриптов для интернет-магазина и mysql база данных, содержащая таблицы для хранения информации, которая используется в работе магазина: данные о товарах, пользователях, заказах и т. д.

2. При проектировании и разработке php — скриптов автор старался придерживаться архитектуры Model-View-Control. Все классы, спроектированные в работе, разделены на 3 типа: классы — контроллеры, граничные классы - формы и классы сущности, непосредственно взаимодействующие с базой данных. Такое типовое построение скриптов значительно упростило разработку приложения, а также позволяет в дальнейшем «наложить» на разработанный функционал интернет-магазина практически любой дизайн, так как логика работы скриптов отделена от интерфейса пользователя [1]. Отход от архитектуры Model-View-Control имел место только в случае высоких накладных расходов на создание объектов — сущностей. В этом случае, нагрузка на сервер баз данных или веб-сервер несоразмерна с решаемой в скрипте задачей.

3. Для создания проекта интернет-магазина использовался язык UML в редакции 1.1. С помощью свободно-распространяемой UML - среды разработки «Astah comunity» были разработаны канонические диаграммы [5]: диаграмма вариантов использования, комплект диаграмм последовательности, комплект диаграмм классов и диаграммы деятельности. Для создания диаграмм использовалось описание потоков событий. Методы объектно-ориентированного анализа и проектирования значительно ускорили процесс проектирования интернет-приложения. Единственная сложность, с которой столкнулся разработчик — это то, что при проектировании диаграмм последовательности необходимо точно знать возможности используемой для реализации технологии. Так, первоначально, автор полагал хранить корзину в XML файле на стороне клиента, что оказалось затруднительно. Выходом из положения стало использование cookies.

Для проектирования базы данных интернет магазина, сначала были выделены сущности, участвующие в работе интернет-магазина, а затем составлена ERD-диаграмма[8]. Web-система администрирования баз данных phpmyAdmin позволяет сразу превратить ERD-диаграмму в базу данных.

4. Для создания интернет-магазина использовался язык программирования php, отличающийся простотой синтаксиса, большим количеством встроенных функций, разнообразием поддерживаемых типов данных и возможностью использования объектно-ориентированного подхода [3,4,6]. Php скрипты выполняются на web-сервере. Для разработки интернет-магазина использовалась версия языка 5.4.16. В качестве web-сервера использовался Apache2. Простота и надежность отличает совместное использование этих технологий.

5. Для передачи данных на сервер использовались 2 метода: GET и POST. Метод пост POST более безопасен, чем GET, параметры в котором передаются через URL[3].

6. Пароль и имя пользователя к базе данных интернет-магазина хранятся в сессии, которая запускается после аутентификации пользователя. Механизм сессий позволяет сохранять переменные для 1 подключения, которые могут использоваться в ходе всей работы пользователя в рамках одного подключения [6]. Безопасность хранения переменных сессии обеспечивается двумя способами: либо с помощью cookies, либо с помощью параметров командной строки.

7. В работе также используется механизм cookies — который позволяет сохранять на стороне клиента данные. Coookies хранят информацию о состоянии корзины покупателя. Время жизни cookies может быть задано при их создании. В работе, время жизни ограничивается рамками 1 подключения пользователя [3].

8. В промышленной практике для хранения данных интернет сайтов очень часто используются базы данных. В связке с apache и php традиционно используется сервер баз данных mysql. Благодаря широким возможностям соответствующей библиотеки php, работа с базами данных mysql не составляет особых трудностей [4]. Разграничение доступа пользователей к таблицам базы данных интернет магазина осуществляется силами сервера баз данных. Mysql 5 позволяет осуществлять очень тонкую настройку доступа пользователе к данным вплоть до назначения прав доступа к столбцам таблицы [2]. В этом проекте разработчик ограничился назначением прав доступа: на добавление записей в таблицу, на редактирование записей и на удаление записей из таблицы для каждой таблицы базы данных. Всего в проекте создано 4 пользователя: администратор магазина, продавец, покупатель и пользователь, чья учетная запись может просматривать таблицу фактических пользователей магазина.

8. html — формы интернет-магазина имеют самое простое оформление, содержат кнопки, таблицы, поля ввода текста, радио-кнопки и выпадающие списки. Эскизы форм изготовлены с помощью WYSIWYG HTML-редактора BlueGriffon и блокнота [7].

# 6. Литература

1. ASP.NET MVC Framework / Г. И. Магдануров, В. А. Юнев. —СПб.: БХВ-Петербург, 2010. — 320 с.: ил. — (Профессиональное программирование).

2. MySQL. Руководство администратора.:Пер. с англ. -М.: Издательский дом «Вильямс»,2005.-624 с.

3. Колисниченко Д.Н. Самоучитель PHP 5. - Издпние 3-е — СПб.: Наука и Техника, 2006. - 576 с.:ил.

4. Кузнецов М.В., Симдянов И.В. MySQL на примерах. -Спб.:БХВ-Петербург, 2007.-597 с.:ил.

5. Маслов А.В., Исаков В.В. Практикум про проектированию информационных систем в экономике: учебное пособие. - Томск: Изд-во Томского политехнического университета, 2009. -193.

6. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript.-СПб.: Питер, 2011.-496 с.:ил.

7. Пауэл Томас А.,Уитворт Дэн. Справочник программиста.-М.:АСТ,Мн.: Харвест, 2005.-384 с.

8. Проектирование экономических информационных систем: Учебник Г.Н. Смирнова, АА Сорокин, Ю.Ф.Тельнов; Под ред. Ю.Ф. Тельнова .-М.: Финансы и статистика, 2003 — 512 с.: ил.