

# Интернет-магазин

## Дизайн Решения

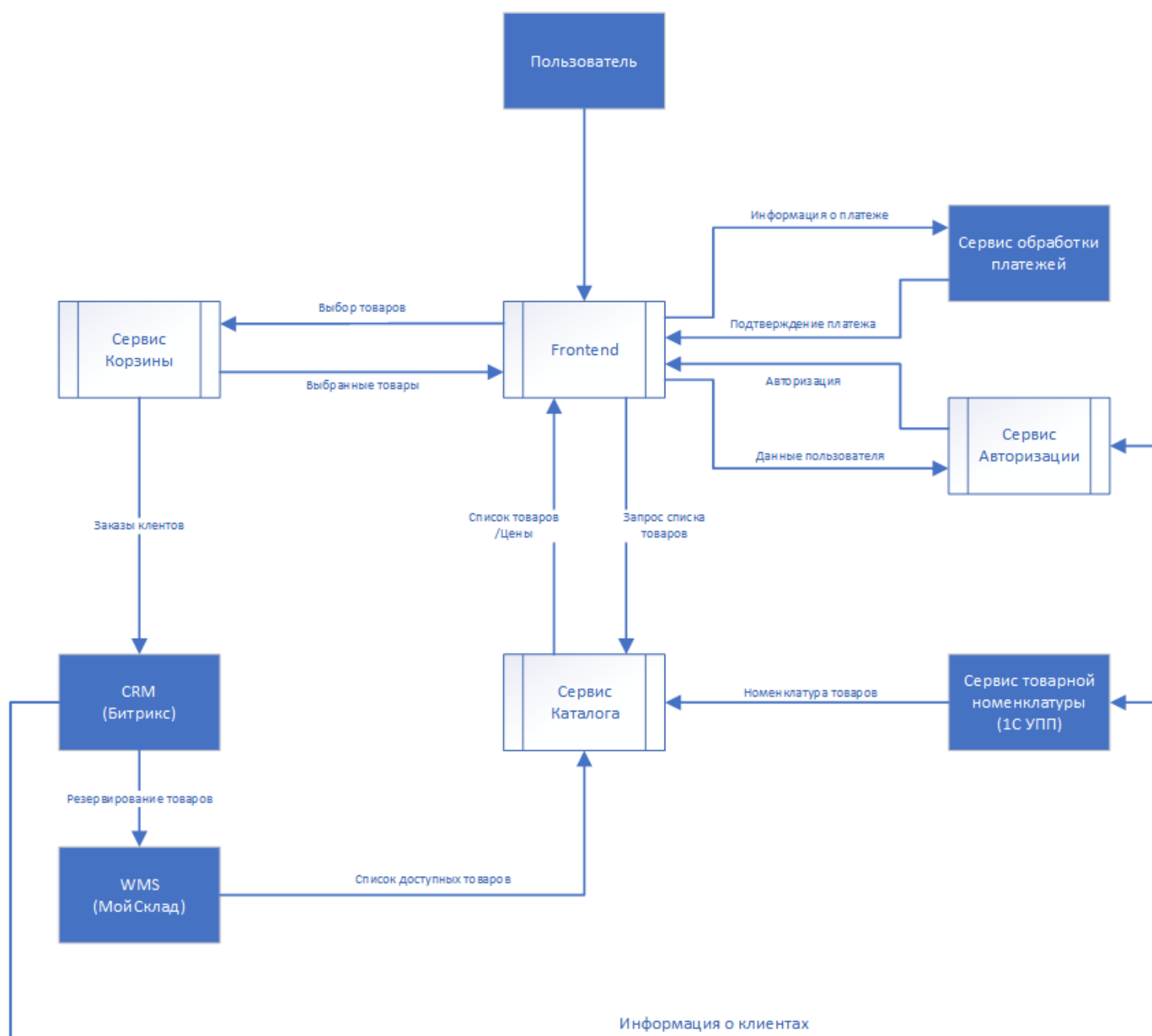
Подготовлено: Кращенко И.А.

Проект: Внедрение интернет-магазина в компании Издательский Дом

# Оглавление

<b>Оглавление</b>	<b>2</b>
<b>Архитектура решения</b>	<b>3</b>
<b>Сервис каталога продуктов</b>	<b>4</b>
Функциональные требования	4
Нефункциональные требования	4
Структура данных	4
Алгоритмы	5
Каталог	5
Поиск по автору	5
Стек используемых технологий	5
<b>Сервис корзины</b>	<b>6</b>
Функциональные требования	6
Нефункциональные требования	6
Структура данных	6
Алгоритмы	6
Стек используемых технологий	7
<b>Интеграция</b>	<b>8</b>
Номенклатура (1С УПП ➡ Сервис каталога)	8
Спецификация OpenAPI (Swagger)	8
Клиенты (Сервис корзины ➡ CRM)	11
Спецификация OpenAPI (Swagger)	11
Каталог (Сервис каталога ➡ Frontend)	14
Спецификация OpenAPI (Swagger)	14
<b>Ролевая модель</b>	<b>16</b>
Роли	16
Матрица доступа	16
<b>Система логирования</b>	<b>17</b>
Уровни логирования	17
Состав логов	17
Схема хранения логов (в днях)	18
<b>Система мониторинга</b>	<b>18</b>
<b>Стратегия резервного копирования</b>	<b>19</b>

# Архитектура решения



# Сервис каталога продуктов

## Функциональные требования

Номер	Тип	Описание
1	Ведение каталога	Система предоставляет возможность создавать, редактировать и удалять карточки товаров: характеристики, цена, наличие.
2	Отзыв	Система предоставляет возможность создавать, редактировать и удалять отзывы о товаре

## Нефункциональные требования

Номер	Тип	Описание
1	Достоверность	Информация в карточках товара должна соответствовать действительным значениям цен, наличия.
2	Доступ к управлению отзывами	Отзыв не может быть изменен и/или удален администратором или менеджером

## Структура данных

Catalog	
 PK	prod_id INT
<hr/>	
	title VARCHAR(100)
	author VARCHAR(50)
	publisher VARCHAR(50)
	year_publish INT

## Алгоритмы

### Каталог

```
SELECT * FROM Catalog;
```

### Поиск по автору

```
SELECT *  
FROM Catalog  
WHERE author = 'pelevin';
```

## Стек используемых технологий

- MySQL

# Сервис корзины

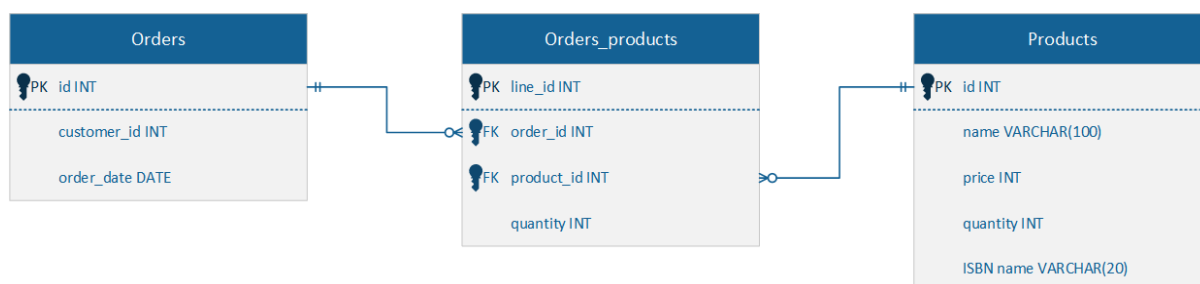
## Функциональные требования

Номер	Тип	Описание
1	Заказ	Система предоставляет возможность добавлять, изменять количество и удалять товары из корзины, которые в дальнейшем сформируют заказ.
2	Заказ на доставку	Система предоставляет возможность создавать, редактировать и удалять заказ на доставку с указанием данных клиента.

## Нефункциональные требования

Номер	Тип	Описание
1	Наличие товара	Нельзя добавить в корзину товар, которого нет в наличии
2	Адрес доставки	Доставка осуществляется в пределах МКАД. Нельзя ввести данные адреса клиента, который находится за пределами МКАД

## Структура данных



## Алгоритмы

### Заказы определенного клиента с товарами

```
SELECT Orders.id, Products.name, Products.price,
Orders_Products.quantity
FROM Orders
JOIN Orders_Products ON Orders_products.order_id = Orders.id
JOIN Products ON Orders_products.product_id = Products.id
```

```
WHERE Orders.customer_id = 12;
```

## Стек используемых технологий

- mySQL

# Интеграция

## Номенклатура (1С УПП ➡ Сервис каталога)

### Спецификация OpenAPI (Swagger)

```
swagger: "2.0"
info:
  description: "Full catalog for a bookstore."
  version: "1.0.0"
  title: "Nomenclature 1C"
  contact:
    email: "test@test.ru"
  license:
    name: "Apache 2.0"
    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
host: "bearshop.ru"
basePath: "/v1"

paths:
  /addBook:
    get:
      tags:
        - "Nomenclature"
      summary: "adds book"
      description: "add books to the catalog"
      operationId: "addBook"
      consumes:
        - "application/json"
      produces:
        - "application/json"
      parameters:
        - name: "prod_id"
          in: "query"
          description: "prod_id for filter"
          required: true
          type: "number"
      responses:
        "200":
          description: "Successful"
          schema:
            type: "array"
            items:
```



```

        $ref: "#/definitions/Book"
    "400":
        description: "Error"
    security:
    - api_key: []

/deleteBook:
    delete:
        tags:
        - "Nomenclature"
        summary: "deletes product"
        description: "Deletes a book from the catalog"
        operationId: "deleteProduct"
        consumes:
        - "application/json"
        produces:
        - "application/json"
        parameters:
        - name: "deleteBookID"
          in: "body"
          description: "Delete product"
          required: true
          schema:
            type: "number"
        responses:
            "200":
                description: "Successful"
                schema:
                    type: "array"
                    items:
                        $ref: "#/definitions/Book"
            "400":
                description: "Error"
        security:
        - api_key: []

securityDefinitions:
    api_key:
        type: "apiKey"
        name: "api_key"
        in: "header"

definitions:
    Book:
        type: "object"
        properties:

```

```
    isbn:
      type: "string"
    name:
      type: "string"
    url:
      type: "string"
    quantity:
      type: "number"
    price:
      type: "number"
  externalDocs:
    description: "Find out more about Swagger"
    url: "http://swagger.io"
```

## Клиенты (Сервис корзины ➔ CRM)

### Спецификация OpenAPI (Swagger)

```
swagger: "2.0"
info:
  description: "Shopping cart for a bookstore."
  version: "1.0.0"
  title: "Shopping cart"
  contact:
    email: "ily20692005@yandex.ru"
  license:
    name: "Apache 2.0"
    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
host: "bearshop.ru"
basePath: "/v1"

paths:
  /shoppingCart:
    get:
      tags:
        - "Shopping-cart"
      summary: "getShopping-cart"
      description: "Get list of Shopping-cart"
      operationId: "getShoppingCart"
      consumes:
        - "application/json"
      produces:
        - "application/json"
      parameters:
        - name: "order_id"
          in: "query"
          description: "Order_id for filter"
          required: false
          type: "number"
      responses:
        "200":
          description: "Successful"
          schema:
            type: "array"
            items:
              $ref: "#/definitions/Book"
        "400":
          description: "Error"
      security:
        - api_key: []
```

```

/createOrder:
  post:
    tags:
      - "Shopping-cart"
    summary: "Adds product"
    description: "Adds an item to the shopping cart"
    operationId: "updateProduct"
    consumes:
      - "application/json"
    produces:
      - "application/json"
    parameters:
      - name: "Product"
        in: "body"
        description: "New product"
        required: true
        schema:
          $ref: "#/definitions/Book"
    responses:
      "200":
        description: "Successful"
        schema:
          type: "array"
          items:
            $ref: "#/definitions/Book"
      "400":
        description: "Error"
    security:
      - api_key: []

/DeleteItemFromOrder:
  delete:
    tags:
      - "Shopping-cart"
    summary: "Deletes product"
    description: "Deletes an item in the shopping cart"
    operationId: "deleteProduct"
    consumes:
      - "application/json"
    produces:
      - "application/json"
    parameters:
      - name: "deleteProductID"
        in: "body"
        description: "Delete product"
        required: true

```

```

    schema:
      type: "number"
  responses:
    "200":
      description: "Successful"
      schema:
        type: "array"
        items:
          $ref: "#/definitions/Book"
    "400":
      description: "Error"
  security:
    - api_key: []

securityDefinitions:
  api_key:
    type: "apiKey"
    name: "api_key"
    in: "header"

definitions:
  Book:
    type: "object"
    properties:
      isbn:
        type: "string"
      name:
        type: "string"
      url:
        type: "string"
      quantity:
        type: "number"
      price:
        type: "number"
externalDocs:
  description: "Find out more about Swagger"
  url: "http://swagger.io"

```

## Каталог (Сервис каталога ➡ Frontend)

### Спецификация OpenAPI (Swagger)

```
swagger: "2.0"
info:
  description: "Full catalog for a bookstore."
  version: "1.0.0"
  title: "Catalog"
  contact:
    email: "test@test.ru"
  license:
    name: "Apache 2.0"
    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
host: "bearshop.ru"
basePath: "/v1"

paths:
  /getBooks:
    get:
      tags:
        - "Catalog"
      summary: "Get books"
      description: "Get list of books from catalog"
      operationId: "getBooks"
      consumes:
        - "application/json"
      produces:
        - "application/json"
      parameters:
        - name: "booksID"
          in: "body"
          description: "books' to add id"
          required: true
          schema:
            type: "number"
      responses:
        "200":
          description: "Successful"
          schema:
            type: "array"
            items:
              $ref: "#/definitions/Book"
        "400":
          description: "Error"
      security:
```

```
- api_key: []

securityDefinitions:
  api_key:
    type: "apiKey"
    name: "api_key"
    in: "header"

definitions:
  Book:
    type: "object"
    properties:
      isbn:
        type: "string"
      name:
        type: "string"
      url:
        type: "string"
      quantity:
        type: "number"
      price:
        type: "number"
externalDocs:
  description: "Find out more about Swagger"
  url: "http://swagger.io"
```

# Ролевая модель

## Роли

- Администратор магазина  
Ведение каталога и карточек товаров  
Подстраховывает менеджера по обработке заказов в непредвиденной ситуации  
Чтение отзывов
- Менеджер по обработке заказа  
Принимает заказ, связывается с клиентом для уточнения деталей,  
Создает заказ на доставку курьеру
- Курьер  
Сверка с каталогом  
Доставка заказа клиенту
- Клиент  
Исследование каталога ,создание заказа, оплата, возможность оставить отзыв

## Матрица доступа

Роль/Объект	Каталог	Товар	Заказ	Заказ на доставку	Отзыв
Администратор	Create Read Update Delete	Create Read Update Delete	Read Update Delete	Read Update Delete	Read
Менеджер по обработке заказов	Read	Read	Update	Create Read Update	Read
Курьер	Read	Read		Read Update	
Клиент	Read	Read	Create Read Update		Create Read Update Delete



# Система логирования

## Уровни логирования

Уровень	Описание	Окружение
Debug	Максимальная детализация любого действия	DEV TEST
Info	Информация о совершаемых действиях и операциях	DEV TEST PROD
Warning	Непредусмотренные обстоятельства: поведение системы, параметры вызова, входящие данные...	DEV TEST STAGE PROD
Error	Ошибка системы, при которой возможно продолжить работу	DEV TEST STAGE PROD DAILY
Fatal	Ошибка системы, при которой с ней НЕ возможно продолжить работу	DEV TEST STAGE PROD DAILY

## Состав логов

- Дата/Время (серверное)
- Уровень лога
- Место возникновения (сервис/модуль/функция)
- Login пользователя
- Описание действий пользователя/системы

## Схема хранения логов (в днях)

	DEV	TEST	STAGE	PROD	DAILY
Debug	3	3	2	1	60
Info	3	3	2	1	180
Warning	3	3	2	1	180
Error	4	4	2	1	180
Fatal	4	4	2	1	180

## Система мониторинга

Сервис/Метрика	Frontend	Сервис корзины	Сервис каталога	Сервис авторизации
Дисковое пространство	10%	10%	10%	10%
CPU	80%	80%	80%	80%
RAM	80%	80%	80%	80%
SELECT (БД)	---	2 сек.	3 сек.	1-2 сек.
GET (HTTP)	1 сек.	5 сек.	1,5 сек.	0,5 сек.

## Стратегия резервного копирования

окружени е	DEV	TEST	STAGE	DAILY	PROD
Назначен ие	Разработка	Тестировани	Подгото вка релиза	Анализ инцидент ов	Промышлен ная эксплуатаци я
Обновле ние с PROD	По запросу	По запросу	По запросу	Ежедневн о	-
Объекты резервир ования	Базы данных сервисов	Базы данных сервисов	Базы данных сервисо в	Базы данных сервисов	Базы данных сервисов
Порядок резервно го копирова ния	Ежедневно 1 раз в сутки полную копию (ночью)  Полная копия каждые 14 дней (размер спринта в ночь)	Ежедневно 1 раз в сутки полную копию (во время минимальной нагрузки - ночью)	-	-	Ежедневная полная копия 1 раз в сутки (ночью пн-вс, днем вс-пн чтобы не пересекатьс я с недельным копировани ем )  Полная резервная копия 1 раз в неделю (ночью с воскресень я на понедельни к)  Дифференц иальная копия - 1 раз в два часа  Полная резервная

					копия журнала транзакций - раз в 15 минут (исключить ночное время: с 7:00 до 23:00)
Сроки хранения резервны х копий	Ежедневна я копия - 2 месяца (60 копий каждой базы)  Полная копия каждые 14 дней - 2 месяца (4 копии)	Ежедневная копия - 2 недели (14 копий каждой базы)	-	-	Ежегодная полная копия - 1 раз 31.12 - 50 лет.  Еженедельн ая полная копия - 6 месяцев (24 копии каждой базы)  Ежедневная копия полная - 3 месяца (90 копий каждой базы)  Дифференц иальная копия - 7 дней (168 копий каждой базы)  копия журнала транзакций - 2 суток (122 копии для каждой базы)