

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

**Согласовано**

Доцент департамента  
программной инженерии  
факультета компьютерных наук  
канд. техн. наук

\_\_\_\_\_ Александров Д.В.  
"    "    \_\_\_\_\_ 2018 г

**Утверждаю**

Академический руководитель  
образовательной программы  
«Программная инженерия»  
профессор департамента программной  
инженерии канд. техн. наук

\_\_\_\_\_ Шилов В. В.  
"    "    \_\_\_\_\_ 2018 г

**Клиент-Серверное Веб-Приложение для Управления Скидками в  
Розничных Сетях**

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.506900 81 01-1

Студент группы БПИ 151 НИУ ВШЭ  
\_\_\_\_\_ Суровцев М.А.  
"    "    \_\_\_\_\_ 2018 г

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

2018

УТВЕРЖДЕНО  
RU.17701729.506900 81 01-1

## Клиент-Серверное Веб-Приложение для Управления Скидками в Розничных Сетях

Пояснительная записка

RU.17701729.506900 81 01-1

Листов 27

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2018

## 1. Аннотация

В данном программном документе приведена пояснительная записка к программе «Клиент-Серверное Веб-Приложение для Управления Скидками в Розничных Сетях». В данном программном документе, в разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка. В разделе «Назначение и область применения» указано функциональное назначение программы и краткая характеристика области применения программы. В данном программном документе, в разделе «Технические характеристики» содержатся следующие подразделы:

- постановка задачи на разработку программы
- описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами
- описание и обоснование выбора состава технических и программных средств

Так же к документу прикреплены приложения. Настоящий документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [5];
2. ГОСТ 19.102-77 Стадии разработки [6];
3. ГОСТ 19.103-77 Обозначения программ и программных документов [7];
4. ГОСТ 19.104-78 Основные надписи [8];
5. ГОСТ 19.106-78 Требования к программным документам [9];
6. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [10].

Изменения к данной Пояснительной записке оформляются согласно ГОСТ 19.603-78 [11]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

# Содержание

<b>1</b>	<b>Аннотация</b>	<b>1</b>
<b>2</b>	<b>Введение</b>	<b>3</b>
2.1	Наименование программы . . . . .	3
2.2	Краткая характеристика . . . . .	3
<b>3</b>	<b>Назначение разработки</b>	<b>4</b>
3.1	Функциональное назначение . . . . .	4
3.2	Эксплуатационное назначение . . . . .	4
<b>4</b>	<b>Технические характеристики</b>	<b>5</b>
4.1	Постановка задачи на разработку программы . . . . .	5
4.1.1	Состав выполняемых функций. Клиентская часть. . . . .	5
4.2	Описание алгоритма и функционирования программы . . . . .	5
4.2.1	Описание базы данных . . . . .	5
4.2.2	Список покупок . . . . .	7
4.2.3	Описание API . . . . .	7
4.2.4	Авторизация . . . . .	13
4.2.5	Клиентское веб-приложение . . . . .	14
4.3	Описание и обоснование выбора метода организации входных и выходных данных	17
4.4	Описание выбора состава технических средств . . . . .	17
4.4.1	Обоснование выбора состава технических средств . . . . .	17
4.4.2	Описание выбора состава программных средств . . . . .	17
4.4.3	Обоснование выбора состава программных средств . . . . .	18
<b>5</b>	<b>Технико-экономические показатели</b>	<b>19</b>
5.1	Предполагаемая потребность . . . . .	19
5.2	Экономические преимущества разработки . . . . .	19
<b>6</b>	<b>Источники, используемые при разработке</b>	<b>20</b>
<b>7</b>	<b>Приложение 1. Терминология</b>	<b>21</b>
7.1	Терминология . . . . .	21

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

<b>8 Приложение 2. Описание классов</b>	<b>22</b>
8.1 Discounts.vue . . . . .	22
8.2 Header.vue . . . . .	22
8.3 Item.vue . . . . .	22
8.4 BellPuzzleLauncher . . . . .	23
8.5 KeyDoorOpener . . . . .	23
8.6 LastPuzzleLogic . . . . .	23
8.7 MainMenu . . . . .	24
8.8 ModChanger . . . . .	24
8.9 NumberChanger . . . . .	24
8.10 OpenDoorAndLoadScene . . . . .	24
8.11 PickUpObject . . . . .	25
8.12 StoryLogic . . . . .	25
8.13 Unscrew . . . . .	25
8.14 VRSlider . . . . .	25

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

## 2. Введение

### 2.1. Наименование программы

Наименование программы на русском: «Клиент-Серверное Веб-Приложение для Управления Скидками в Розничных Сетях».

Наименование на английском: «The Client-Server Web Application for Managing the Products Discounts in Retail Networks».

### 2.2. Краткая характеристика

Клиент-серверное Web приложение для отслеживания скидок и акций в продуктовых магазинах. Основные функциональные возможности приложения: представление каталога акций в удобном и доступном виде, работа со списком покупок, отображение ближайших магазинов на карте.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

### 3. Назначение разработки

#### 3.1. Функциональное назначение

Продукт является приложением “на стыке онлайн и оффлайна”, позволяющим планировать покупки в магазине с учетом актуальных скидок. К функциональным возможностям программы относятся: просмотр списка актуальных акций, составление и редактирование списка покупок, поиск ближайшего магазина на карте, регистрация пользователей.

#### 3.2. Эксплуатационное назначение

Программа предназначена для запуска на персональных компьютерах (веб-версия) и мобильных устройствах (веб-версия). Программа является приложением, позволяющим пользователю искать и подбирать интересующие его товары по сниженной цене, и, таким образом, экономить средства.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

## 4. Технические характеристики

### 4.1. Постановка задачи на разработку программы

Программа должна соответствовать требованиям, представленным в Техническом Зада-  
нии.

Задачи работы (серверная часть приложения):

1. Реализовать REST API:
  - (a) Обработка запросов для работы со списками покупок пользователей
  - (b) Обработка запросов для кроулера
  - (c) Обработка запросов для получения списка актуальных акционных товаров
  - (d) Обработка запросов для авторизации пользователей
2. Спроектировать базу данных для хранения:
  - (a) Акционных товаров
  - (b) Аккаунтов пользователей
  - (c) Списков покупок пользователей

Задачи работы (клиентская часть приложения):

#### 4.1.1. Состав выполняемых функций. Клиентская часть.

1. Реализовать возможность просмотра списка доступных магазинов с акционными то-  
варами
2. Реализовать представление текущих акций для конкретного магазина:
  - (a) В виде общего списка
  - (b) По категориям
3. Реализовать возможность регистрации и авторизации пользователей в системе для  
работы со списком покупок
4. Реализовать создание и удаление списка покупок
5. Реализовать работу со списком покупок:
  - (a) Добавление и удаление товаров магазинов
  - (b) Добавление и удаление пользовательских позиций
  - (c) Рекомендация товаров магазинов на основе пользовательских позиций

### 4.2. Описание алгоритма и функционирования программы

#### 4.2.1. Описание базы данных

Пожалуй, главной сущностью в схеме является «item». Именно в эту таблицу записыва-  
ются товары, собранные кроулером, и из этой таблицы извлекаются данные для отобра-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата



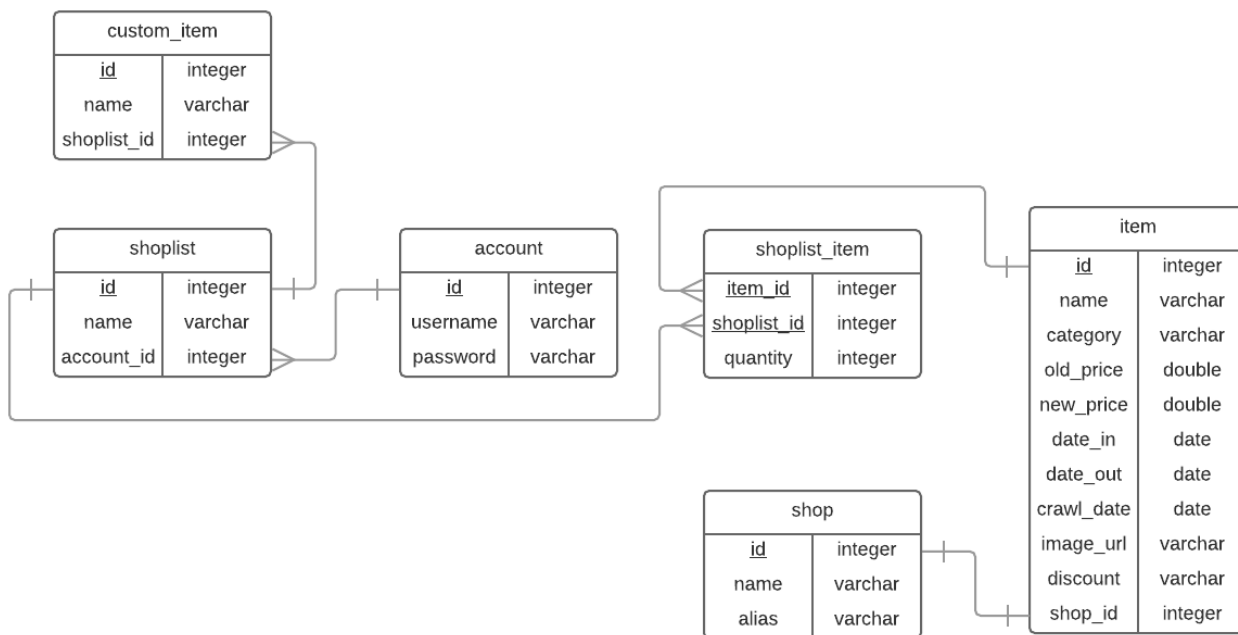


Рис. 1: Схема базы данных

жения в клиентских приложениях. Каждый товар также связан с магазином, которому он принадлежит. (Один магазин может иметь много товаров).

Как видно из Рис. 1, пользователь может иметь несколько списков покупок, которые в свою очередь могут содержать много товаров.

Серверная часть приложения написана на NodeJS и использует ORM Sequelize. Каждая таблица в схеме базы данных имеет свое отражение в виде объекта в программном коде. Отношения (one-to-one, one-to-many и many-to-many) задаются через класс Association. [2]

```

// Модель таблицы account в виде объекта в JS коде
module.exports = function(sequelize, DataTypes) {
  return sequelize.define('account', {
    id: {
      type: DataTypes.INTEGER,
      allowNull: false,
      primaryKey: true,
      autoIncrement: true
    },
    username: {
      type: DataTypes.STRING,
      allowNull: false
    },
    password: {
      type: DataTypes.STRING,

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

```
        allowNull: false
    }
});
};

// Пример отношений между сущностями
Account.hasMany(ShopList); // Один аккаунт имеет много списков покупок
ShopList.belongsTo(Account); // Список покупок принадлежит одному аккаунту
Item.belongsToMany(ShopList, { through: ShopListItem }); // Товар принадлежит
// многим спискам покупок
ShopList.belongsToMany(Item, { through: ShopListItem }); // Список покупок
// принадлежит многим товарам
```

#### 4.2.2.Список покупок

Список покупок имеет следующую структуру:

1. Имя списка покупок (Например, «Завтрак»)
2. Список товаров магазинов (Например, «Молочный коктейль Чудо детки шоколад; клубника 2,5%, 200 мл»)
3. Пользовательские позиции (Например, «Сок»)
4. Итоговая сумма покупок

При добавлении пользовательской позиции в список покупок, система предложит пользователю актуальные товары из магазинов.

Так, например, для позиции «Сок» будут предложены следующие товары, которые пользователь может добавить в список товаров магазинов:

1. Сок Сады Придонья Яблоко зеленое 125мл
2. Сок Добрый Яблочный 2л
3. Десерт Джелео многослойный с соком вишня-персик-яблоко 0,4%, 150 г

Список товаров магазинов и предложенные товары на основе пользовательских позиций для удобства группируются по магазинам. Добавлять товары в список покупок могут только авторизованные пользователи.

#### 4.2.3.Описание API

В этом разделе перечислены все endpoint'ы для реализованного API. Для каждого endpoint'а приведено краткое описание и пример ответа в формате json.

##### Работа со списком акций

/api/shops – получить список всех магазинов

GET-запрос

Пример ответа:

```
[
  {
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

```

    "id": 1,
    "alias": "dixy",
    "name": "Дикси"
  },
  {
    "id": 2,
    "alias": "perekreсток",
    "name": "Перекресток"
  }
]

```

/api/shops – загрузить новые товары на сервер. Этот endpoint нужен для кроулера, чтобы записать собранные акции в базу данных.

POST-запрос

Тело запроса:

```

{
  "name": "Компот Д из персиков, 580 мл",
  "category": "Консервы, соусы",
  "oldPrice": 137,
  "newPrice": 99.99,
  "dateIn": "2018-04-02",
  "dateOut": "2018-04-08",
  "crawlDate": "2018-04-02",
  "condition": "-",
  "image": null,
  "imageUrl": "https://dixy.ru/upload/iblock/fe2/2000183687.jpg",
  "discount": "-27",
}

```

/api/shops/:shopid – получить список товаров для данного магазина

GET-запрос

Параметры URL:

1. shopid – id магазина

Параметры запроса (опционально):

1. category – название категории (по умолчанию возвращаются товары всех категорий)
2. page – номер страницы (по умолчанию 1)

Пример ответа:

/api/shops/1?category=Кулинария,%20заморозка,%20мороженое&page=1

```

{
  "count": 192,
  "rows":
  [

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

```
{
  "id": 565,
  "name": "Мороженое Жемчужина России эскимо миндаль-карамель, 80 г",
  "category": "Кулинария, заморозка, мороженое",
  "oldPrice": 52.9,
  "newPrice": 26.45,
  "dateIn": "2018-03-15",
  "dateOut": "2018-03-28",
  "crawlDate": "2018-03-17",
  "condition": "-",
  "image": null,
  "imageUrl": "https://dixy.ru/upload/iblock/814/2000148579.jpg",
  "discount": "1+1",
  "shopId": 1
},
"numPages": 1
}
```

Здесь:

1. count – общее количество товаров для данного магазина
2. numPages – всего страниц с товарами. Рассчитывается как:

$$\text{Math.ceil}(\text{count}/\text{ITEMS\_PER\_PAGE}), \text{ITEMS\_PER\_PAGE} = 30$$

3. rows – список с товарами

Видно, что сервер отдает данные постранично. Т.е. вместо того, чтобы отдать сразу все 192 товара, сервер возвращает страницы, содержащие по 30 товаров. Поля dateIn и dateOut у объекта item нужны для того, чтобы загружать только актуальные товары из базы. Т.е. выполняется запрос вида:

```
SELECT * FROM item where date_in <= currDate AND date_out >= currDate
```

/api/shops/:shopid/categories – получить список категорий для данного магазина

GET-запрос

Параметры URL:

1. shopid – id магазина

Пример ответа:

["Фермерские продукты", "Хлеб, сладости, снеки", "Овощи, фрукты, грибы", "Товары для животных", "Товары для мам и детей", "Консервы, орехи, соусы", "Макароны,

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

крупы, специи", "Мясо, птица, деликатесы", "Соки, воды, напитки", "Молоко, сыр, яйца", "Авто, дом, сад, кухня", "Наши марки", "Рыба и морепродукты", "Красота, гигиена, бытовая химия", "Кофе, чай, сахар", "Замороженные продукты", "Здоровый выбор"]

### Работа со списком покупок

Все запросы из этого раздела должны быть авторизованы. (см. раздел Авторизация)

/api/shoplist – получить списки покупок для данного пользователя

GET-запрос

Параметры запроса:

1. mode – режим для загрузки списков покупок

(a) preview – списки покупок загружаются в сжатом виде

(b) full – списки покупок загружаются в полном виде

Если параметр не задан, то загружаются списки покупок без товаров

Примеры ответа:

/api/shoplist

```
[
  {
    "id": 17,
    "name": "Купить"
  },
  {
    "id": 18,
    "name": "Чё купить"
  },
  {
    "id": 19,
    "name": "Zzz"
  }
]
```

/api/shoplist?mode=preview

```
[
  {
    "id": 17,
    "name": "Купить",
    "items": [
      "Мыло Солнышко хозяйственное с ароматом лимона 140г",
      "Крем-мыло жидкое Особая серия овсяное молочко, 1000 г"
    ],
    "customItems": [
      "Мыло"
    ]
  }
]
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

```
]
},
{
  "id": 18,
  "name": "Чё купить",
  "items": [
    "Мешок для мусора Фрекен БОК 35л 15шт",
    "Сок Добрый Яблочный 2л",
    "Сосиски Молочные Велком, 530 г",
    "Нектар Добрый апельсиновый; мультифруктовый; персик-яблоко; груша; виноград-гран
  ],
  "customItems": [
    "сок",
    "колбаса"
  ]
}
]
```

/api/shoplist?mode=full

```
[
  {
    "id": 17,
    "name": "Купить",
    "items": [
      {
        "id": 294,
        "name": "Мыло Солнышко хозяйственное с ароматом лимона 140г",
        "category": "Красота, гигиена, бытовая химия",
        "oldPrice": 0,
        "newPrice": 27.9,
        "dateIn": "2018-03-30",
        "dateOut": "2018-04-20",
        "crawlDate": "2018-03-30",
        "condition": null,
        "image": null,
        "imageUrl": "https://perekrestok.ru/src/product.file/list/image/21/67/16721.jpeg",
        "discount": null,
        "shop": {
          "id": 2,
          "alias": "perekrestok",
          "name": "Перекресток"
        }
      },
      {
        "id": 73880,
        "name": "Крем-мыло жидкое Особая серия овсяное молочко, 1000 г",
        "category": "Непродовольственные товары",

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

```
"oldPrice": 89.9,
"newPrice": 44.95,
"dateIn": "2018-04-02",
"dateOut": "2018-04-08",
"crawlDate": "2018-04-02",
"condition": "-",
"image": null,
"imageUrl": "https://dixy.ru/upload/iblock/267/2000267119.jpg",
"discount": "1+1",
"shop": {
  "id": 1,
  "alias": "dixy",
  "name": "Дикси"
}
},
"customItems": [
  {
    "id": 24,
    "name": "Мыло",
    "shoplistId": 17,
    "matchingItems": [
      {
        "id": 187,
        "name": "Крем-мыло жидкое Dove Красота и уход 250мл",
        "category": "Красота, гигиена, бытовая химия",
        "oldPrice": 0,
        "newPrice": 259,
        "dateIn": "2018-03-30",
        "dateOut": "2018-04-20",
        "crawlDate": "2018-03-30",
        "condition": null,
        "image": null,
        "imageUrl": "https://perekrestok.ru/src/product.file/list/image/74/74/7474.jpeg",
        "discount": null,
        "shop": {
          "id": 2,
          "alias": "perekrestok",
          "name": "Перекресток"
        }
      }
    ]
  }
]
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

В этом примере видно, что помимо списка `items`, соответствующего товарам магазинов, также есть список `customItems`, который представляет пользовательские позиции. У каждой пользовательской позиции есть список `matchingItems`, который содержит товары магазинов, удовлетворяющих данной позиции. Также у каждого товара есть поле `shop`, с помощью которого можно определить, какому магазину принадлежит данный товар, чтобы, например, сгруппировать товары в списке покупок по магазинам.

#### 4.2.4. Авторизация

Для доступа к списку покупок пользователю необходимо авторизоваться. Авторизация на сервере выполняется следующим образом:

1. Пользователь авторизуется в системе, вводя свои логин и пароль
2. Если данные верны, система возвращает токен
3. Полученный токен клиентское приложение добавляет в заголовок каждого защищенного запроса

Токен имеет формат JWT и представлен в следующем виде:

xxxxxx.yyyyyy.zzzzzz

где

xxxxxx – Header (заголовок). В нем содержится метаданные, такая как алгоритм шифрования и тип токена (как правило это JWT)

yyyyyy – Payload – здесь содержится основная информация, в частности время создания токена, имя и id пользователя, получившего токен.

zzzzzz – Signature – сигнатура токена. Позволяет проверить, что токен не был подменен и является валидным. Получается в результате применения SHA256 функции: `SHA256(Header, Payload, Secret key)`, где Secret key – секретный ключ, хранящийся на сервере.

Каждая часть токена закодирована в формат Base64.

Для авторизации в API есть два endpoint'a

1. `/auth/register` – для регистрации в системе
2. `/auth/login` – для входа в систему

В обоих случаях в теле запроса передается следующая информация:

```
{
  "username": "Имя пользователя",
  "password": "Пароль"
}
```

В случае регистрации возвращается статус 200, если регистрация прошла успешно или статус 500, если пользователь ввел короткое имя или пароль, либо такое имя пользователя уже существует.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата



В случае успешного логина возвращается токен. Клиентские приложения должны сохранить его, чтобы в следующий раз отправлять авторизованные запросы. Например, в реализованном веб-приложении такой токен сохраняется в локальном хранилище (localStorage). [4]

#### 4.2.5. Клиентское веб-приложение

Реализованное веб-приложение является SPA (Single Page Application). Таким образом, все данные загружаются с сервера асинхронно с использованием технологии AJAX и модификацией DOM-дерева, контент страницы изменяется динамически.

Приложение написано с использованием JavaScript-фреймворка VueJS и включает следующие компоненты:

1. App.vue – главная компонента приложения, являющаяся родительским контейнером для остальных компонент.
2. Discounts.vue – компонента для отображения списка акций
3. Header.vue – компонента для навигации по сайту
4. ItemSmall.vue – компонента для отображения товара в списке покупок и в рекомендациях
5. Item.vue – компонента для отображения товара в списке акций
6. Login.vue – компонента для отображения формы входа в систему
7. Register.vue – компонента для отображения формы регистрации в системе
8. ShopListCard.vue – компонента для отображения карточки «превью» списка покупок
9. ShopListPreview.vue – компонента-превью для списков покупок
10. ShopList.vue – компонента для отображения списка покупок пользователя
11. ShopListCustomItem.vue – компонента для отображения пользовательской позиции

Компоненты располагаются в приложении иерархически и могут быть вложены (nested components). [3]

Взаимодействие между компонентами происходит через т.н. props (для передачи данных от родительской компоненты дочерней) и через events (для передачи данных от дочерней компоненты родительской). [3]

```
<!-- Объявление компоненты ShopListCard внутри ShopListPreview и  
      передача ей объекта shoplist через props -->
```

```
<shoplist-card :shoplist="shoplist"></shoplist-card>
```

```
<!-- Объявление компоненты ItemSmall внутри ShopListCustomItem и  
      обработка события, исходящего из дочерней компоненты -->
```

```
<item-small v-on:addItem="addItem($event)" >
```

```
</item-small>
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

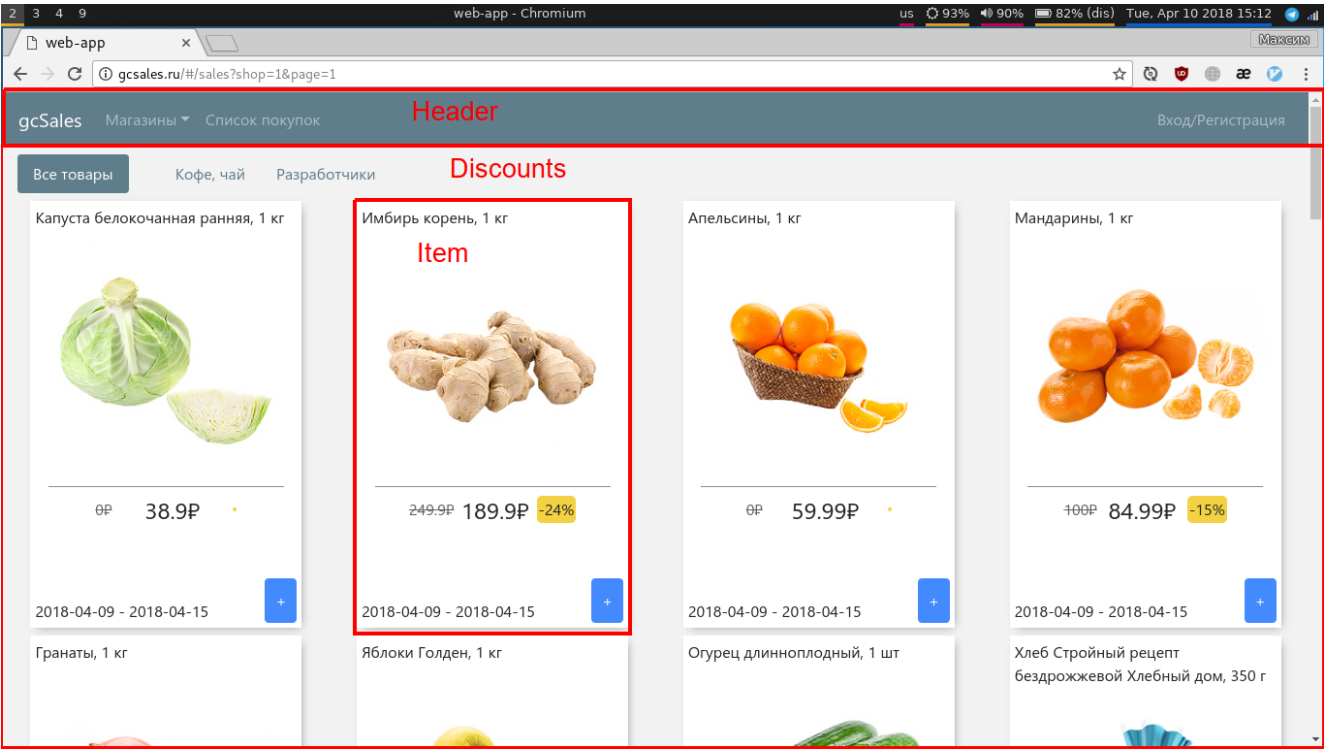


Рис. 2: Компоненты списка акций

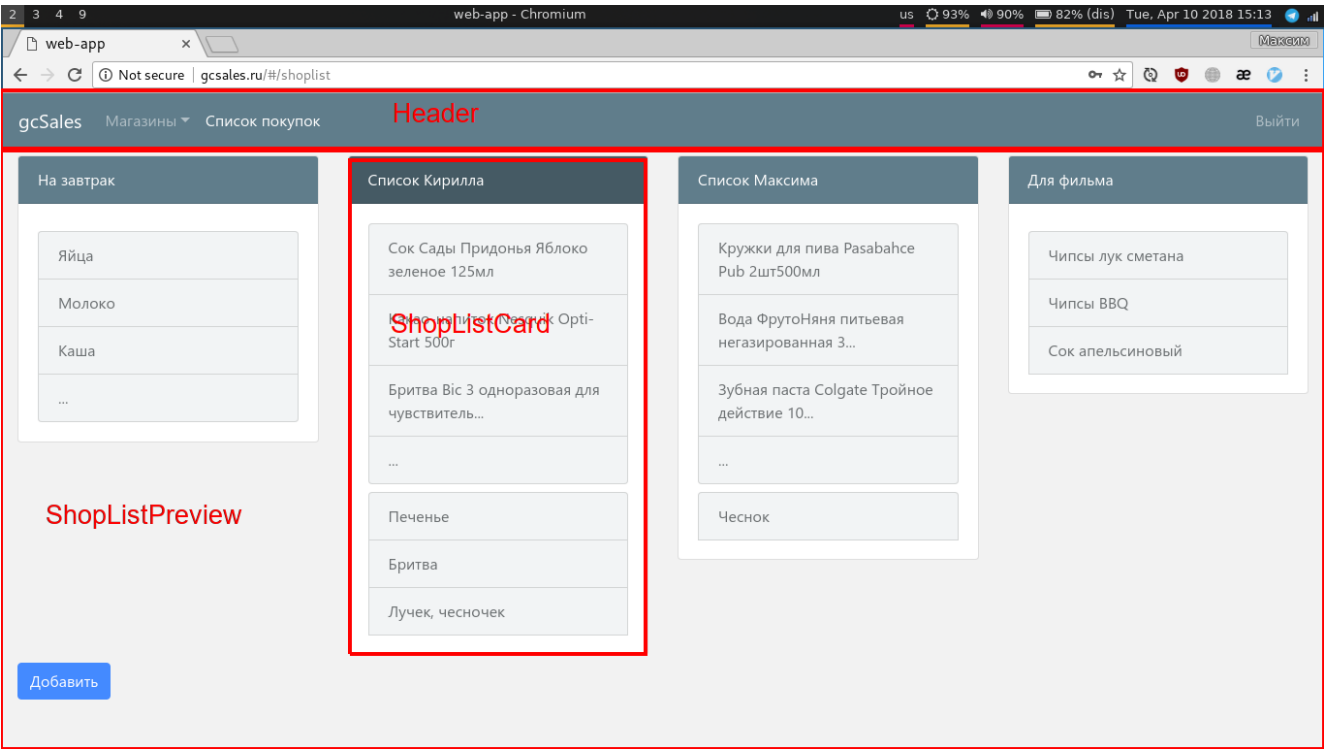


Рис. 3: Компоненты превью списков покупок

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

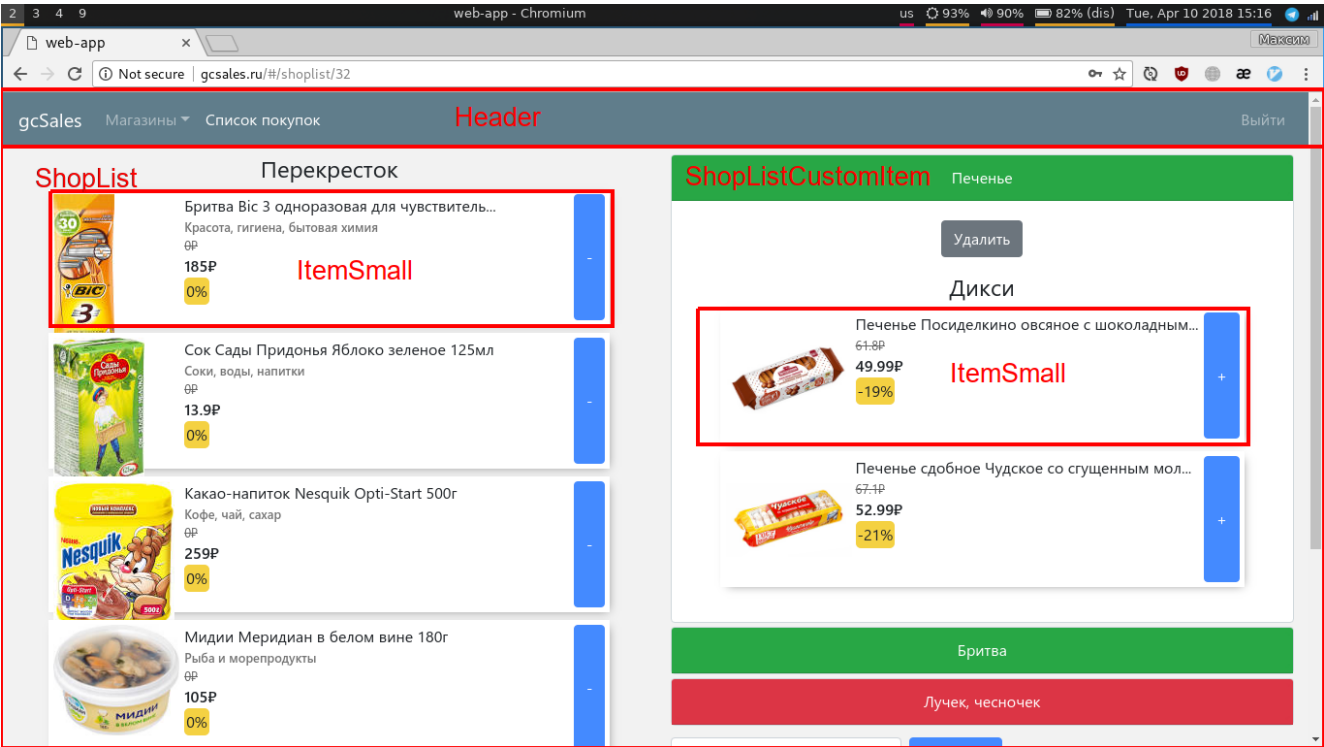


Рис. 4: Компоненты списка покупок

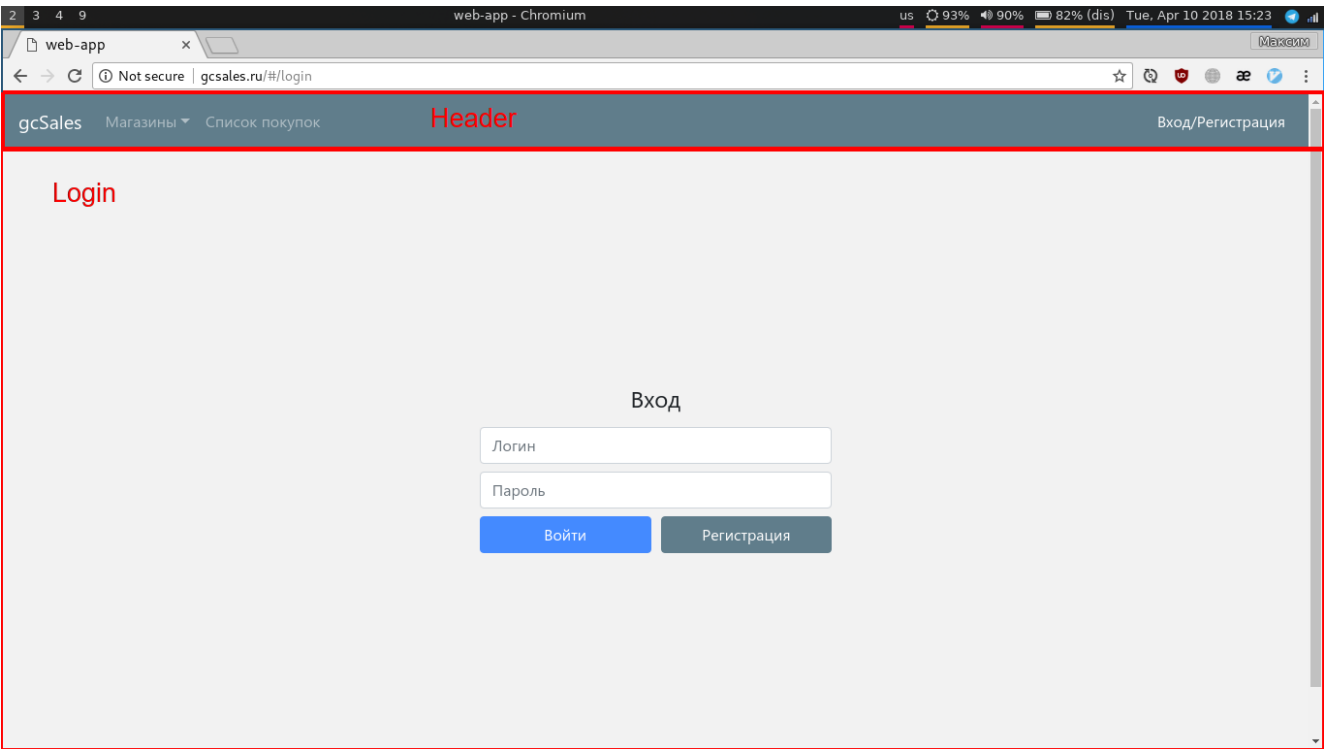


Рис. 5: Компоненты авторизации

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

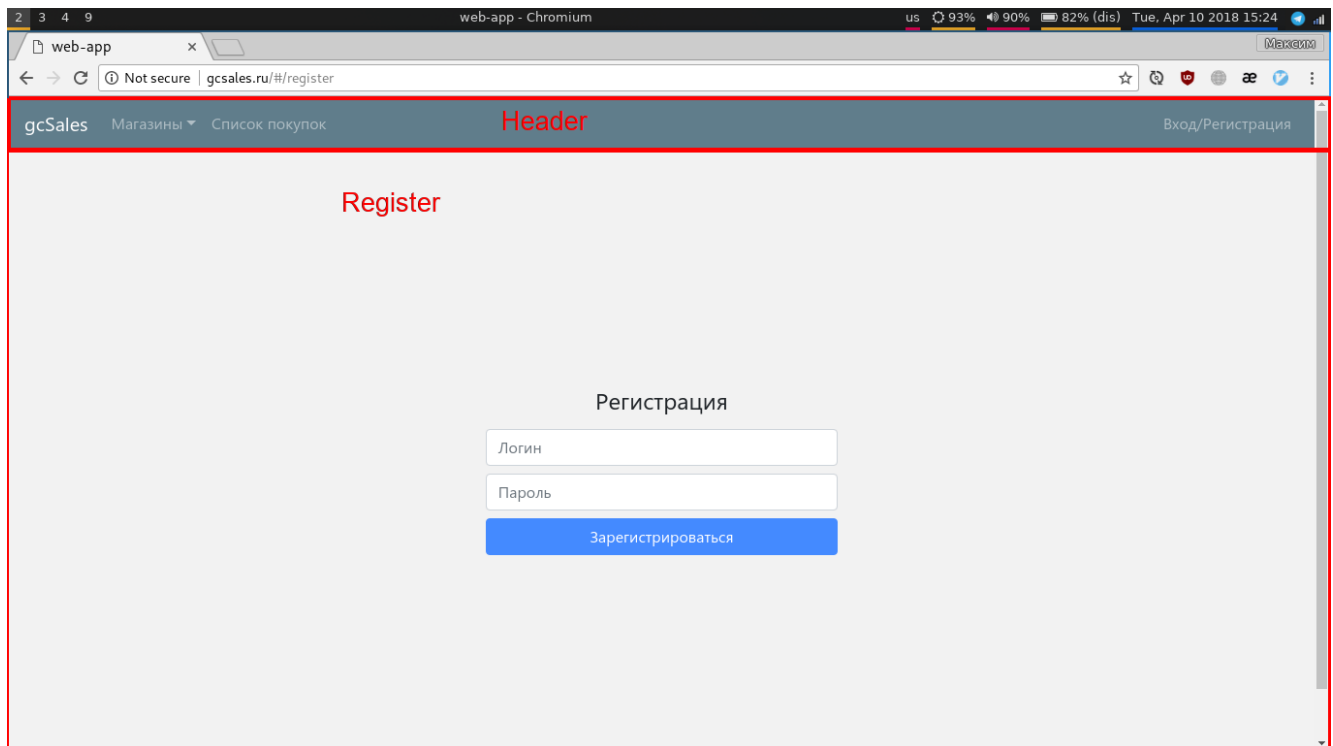


Рис. 6: Компоненты регистрации

#### 4.3. Описание и обоснование выбора метода организации входных и выходных данных

Клиентские приложения и сервер обмениваются данными в формате JSON.

#### 4.4. Описание выбора состава технических средств

Для обеспечения работоспособности программы требуются следующие технические средства:

1. ПК или мобильное устройство
2. Доступ в интернет

##### 4.4.1. Обоснование выбора состава технических средств

Доступ в интернет необходим, т.к. веб-приложение размещено на сетевом ресурсе.

##### 4.4.2. Описание выбора состава программных средств

Для запуска и работы программы требуется ПК или мобильное устройство со следующим предустановленным программным обеспечением:

1. Интернет-браузер Google Chrome версии 49 и выше или
2. Интернет-браузер Firefox версии 44 и выше

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

## 4.4.3. Обоснование выбора состава программных средств

Выбор состава программных средств обоснован минимальными требованиями фреймворка VueJS и библиотеки Bootstrap. [1]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

## 5. Техничко-экономические показатели

### 5.1. Предполагаемая потребность

“Клиент-Серверное Android-Приложение для Управления Скидками в Розничных Сетях” может быть использована в потребительской сфере. Программу могут использовать как люди, нуждающиеся в экономии средств посредством покупки более дешёвых товаров, так и люди, желающие эффективно и быстро работать со списками покупок.

### 5.2. Экономические преимущества разработки

На момент принятия решения о написании данного продукта во Всемирной сети Интернет был лишь 1 аналог: приложение “Едадил”. Преимущества данной разработки перед конкурентом:

1. Лёгкий, не перегруженный излишним функционалом дизайн
2. Лучший сбор картинок (при помощи кроулинга)
3. Возможность создания неограниченного количества списков покупок с разными названиями
4. Возможность добавления в списки покупок товаров, которых нет в наличии в магазинах

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

## 6. Источники, используемые при разработке

### Список литературы

- [1] *Can I use... Support tables for HTML5, CSS3, etc.* 7 апр. 2018. URL: <https://caniuse.com/#search=bootstrap>.
- [2] *Manual | Sequelize | The node.js ORM for PostgreSQL, MySQL, SQLite and MSSQL.* 5 марта 2018. URL: <http://docs.sequelizejs.com>.
- [3] *Vue.js.* 10 февр. 2018. URL: <https://vuejs.org/>.
- [4] *Window.localStorage - Интерфейсы веб API | MDN.* 13 февр. 2018. URL: <https://developer.mozilla.org/ru/docs/Web/API/Window/localStorage>.
- [5] Единая Система Программной Документации. *ГОСТ 19.101-77 Виды программ и программных документов.* ИПК Издательство стандартов, 2001.
- [6] Единая Система Программной Документации. *ГОСТ 19.102-77 Стадии разработки.* ИПК Издательство стандартов, 2001.
- [7] Единая Система Программной Документации. *ГОСТ 19.103-77 Обозначения программ и программных документов.* ИПК Издательство стандартов, 2001.
- [8] Единая Система Программной Документации. *ГОСТ 19.104-78 Основные надписи.* ИПК Издательство стандартов, 2001.
- [9] Единая Система Программной Документации. *ГОСТ 19.106-78 Требования к программным документам.* ИПК Издательство стандартов, 2001.
- [10] Единая Система Программной Документации. *ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению.* ИПК Издательство стандартов, 2001.
- [11] Единая Система Программной Документации. *ГОСТ 19.603-78 Общие правила внесения изменений.* ИПК Издательство стандартов, 2001.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

## 7. Приложение 1. Терминология

### 7.1. Терминология

**REST (от англ. Representational State Transfer)** – способ сетевого взаимодействия. REST API определяет набор функций, к которым разработчики могут совершать запросы и получать ответы. Взаимодействие происходит по протоколу HTTP. Преимуществом такого подхода является широкое распространение протокола HTTP, поэтому REST API можно использовать практически из любого языка программирования.

**REST endpoint** – URL, по которому выполняется запрос к REST API.

**SPA (Single Page Application)** – это веб-приложение или веб-сайт, использующий единственный HTML-документ как оболочку для всех веб-страниц и организующий взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JavaScript, обычно посредством AJAX.

**AJAX** – подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

**DOM** – это независимый от платформы и языка программный интерфейс, позволяющий программам и скриптам получить доступ к содержимому HTML-, XHTML- и XML-документов, а также изменять содержимое, структуру и оформление таких документов.

**ORM (Objec-Relational mapping)** – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата



## 8. Приложение 2. Описание классов

Веб-приложение

### 8.1. Discounts.vue

Поля:

```
data: function() {  
  return {  
    categories: [],  
    items: [],  
    shoplists: [],  
    numPages: 0,  
    user: auth.user  
  }  
}
```

Компонента для отображения списка текущих акций для разных магазинов.

### 8.2. Header.vue

Поля:

```
data: function() {  
  return {  
    shops: [],  
    user: auth.user  
  }  
}
```

Компонента для отображения меню навигации вверху страницы.

### 8.3. Item.vue

Поля:

```
data: function() {  
  return {  
    maxNameLength: 80,  
    showImage: true,  
    user: auth.user  
  }  
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

```
};  
},
```

Компонента для отображения акционного товара.

#### 8.4. BellPuzzleLauncher

Поля:

```
public Transform handMountingPosition;  
private bool hasBell = false;  
private float angle;  
public Transform vrCam;  
public GameObject bellPuzzle;  
public Transform firstToComplete, secondToComplete;  
private bool firstPut = false, secondPut = false;
```

Прикрепляется к невидимому кубу с коллайдером, который при столкновении с персонажем и наведении им указателя на место головоломки вешает колокольчик из руки, и, если все 5 на месте, запускает головоломку.

#### 8.5. KeyDoorOpener

Поля:

```
public GameObject door, final;  
private static bool opened = false;  
private float timeToOpen = 1F;  
public Transform handMountingPosition;
```

Класс, отвечающий за открытие двери когда в руке у игрока находится ключ.

#### 8.6. LastPuzzleLogic

Поля:

```
char[] numbers;  
bool solved;  
float step;  
string key = "4118956";  
public GameObject Key;
```

Класс, отвечающий за поведение последней головоломки с ключем на книжном шкафу.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

## 8.7. MainMenu

Поля:

```
public GameObject character, story;
```

Класс главного меню, прикрепляется к двум лэйблам «PLAY» и «EXIT».

## 8.8. ModChanger

Поля:

```
public GameObject modeChooser;  
public static bool vrModeEnabled = false;  
private bool checked_ = false;
```

Класс, отвечающий за выбор режима (VR Mode или Normal Mode)

## 8.9. NumberChanger

Поля:

```
int n;  
public float timeDelay = 1F;  
float currentTime = 0F;  
bool gazedAt = false;
```

Класс, отвечающий за изменение каждой цифры в последней головоломке.

## 8.10. OpenDoorAndLoadScene

Поля:

```
public GameObject door;  
public static bool opened = false;  
public float timeToOpen = 0.6F;
```

Класс, отвечающий за открытие двери с TransitScene для входа в дом.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

### 8.11. PickUpObject

Поля:

```
public Vector3 handPosition;  
public Vector3 handRotation;  
Vector3 oldScale;  
public float angle1 = 275F, angle2 = 303F;  
public Transform handMountingPosition;  
public Transform vrCam;  
private bool tilted;  
bool pickedUp = false;
```

Класс, отвечающий за перемещение объекта с места, где он лежал, в руку игроку.

### 8.12. StoryLogic

Поля:

```
public GameObject character, vrCam, guide, tryIt, glhf, stone, constraint;  
private bool freezeMove = true, thrown = false, shown = false;
```

Класс, отвечающий за последовательность показа приветственных текстов и обучающего фрагмента вначале.

### 8.13. Unscrew

Поля:

```
bool unscrewed = false, hasScrewDriver = false;  
public Transform handMountingPosition;
```

Класс, отвечающий за откручивание колокольчика со стены при помощи отвертки в руке игрока.

### 8.14. VRSlider

Поля:

```
public float fillTime = 2f;  
public int Scene = 1;  
private Slider mySlider;  
private float timer;  
private bool gazedAt;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

private Coroutine fillBarRoutine;

Класс, отвечающий за анимацию слайдера - индикатора внизу экрана.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ докум.	Входящий № сопроводительного докум. и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.506900 81 01-1				
Инв. №подл.	Подп. и дата	Взам. инв. №	Инв. №дубл.	Подп. и дата