

**Секция «Математическое и программное обеспечение  
информационно-исследовательских систем и ресурсов»**

**научный руководитель – Воронова Лилия Ивановна, доктор физ.-мат. наук, профессор**

**РАЗРАБОТКА ПОДСИСТЕМЫ ПОДБОРА  
СОПУТСТВУЮЩИХ ТОВАРОВ ПРИ ПОКУПКЕ  
В ИНТЕРНЕТ-МАГАЗИНЕ**

Зверев П.Д., Шукенбаев А.Б., Шукенбаева Н.Ш.

*Московский институт государственного  
и корпоративного управления,*

*Российский государственный гуманитарный университет,  
Москва, e-mail: shukenbaev@mail.ru*

Популярность интернет-магазинов неуклонно растет. Основная их задача – обеспечить высокий процент покупок. Для этого часто используют рекомендательные системы. Система рекомендаций сравнивает профиль пользователя с какой-то справочной информацией и старается предсказать «рейтинг», который пользователь даст объекту, над которым он еще даже не думал. Работа рекомендательных систем видна в онлайн магазинах Amazon, Last.fm, Pandora, Google Play и др. Основная задача системы – предложить товары на базе купленных или просмотренных, т.е. рекомендательная система должна предлагать некоторый упорядоченный список товаров, основываясь на предыстории и предпочтениях покупателя.

Целью работы является разработка подсистемы подбора сопутствующих товаров при покупке в интернет-магазине ([www.miko.digitalfashion.ru](http://www.miko.digitalfashion.ru)). Основная задача подсистемы – предложить новые товары на базе купленных-просмотренных. Такая подсистема позволит увеличить ассортимент и продажу товаров и в конечном итоге прибыль в магазине. Для выполнения поставленной цели необходимо было выбрать подходящий метод и разработать приложения. В результате проведенного анализа и сравнения (рассмотрены методы байесовских сетей, гибридные модели, латентные модели и др.) был выбран метод кластерного анализа, на основе которого разрабатывалась подсистема. Для языка Python и в частности для Django используется приложение Django-recommends, которое основывается на коллаборативной фильтрации по рейтингу и использует алгоритм метода рас-

чета сходства и рекомендаций. Метод в качестве входных данных использует список голосов для объекта и возвращает словарь сходств для каждого объекта. Система рекомендаций тесно взаимодействует со структурой построения приложения. Невозможно реализовать систему и правильно настроить ее без создания базовой функциональности магазина.

Для решения поставленных задач были выбраны: операционная система Mac OS X 10.10; СУБД MySQL; язык программирования Python со свободным фреймворком для веб-приложений Django; кроссплатформенный веб-сервер Apache с использованием модуля mod\_wsgi. Дополнительным языком программирования выбран JavaScript, а для взаимодействия JavaScript и HTML использовалась библиотека jQuery.

На рис. 1 представлена ER-модель разрабатываемой подсистемы.

В ходе проектирования и реализации были созданы приложения, состоящие из нескольких файлов, отличающихся своей функциональностью: Например:

- models.py – предназначен для отображения информации о данных.
- admin.py – использует мета-данные модели для представления интерфейса для работы с содержимым сайта.
- test.py. Django предоставляет возможность тестировать код, используя пакет тестов.
- templatetags. Django содержит встроенные теги и фильтры, но в некоторых приложениях может собственный функционал.
- views.py. Представление – это функция Python, которая принимает web-запрос и возвращает web-ответ (HTML содержимое, массив данных или XML файл).
- widgets.py. Виджеты – это представление поля в виде HTML кода, который обеспечивает извлечение данных из POST/GET запросов для расширения базового функционала типа данных как в административной панели, так и на сайте.
- пользовательский файл или класс – для нестандартных задач

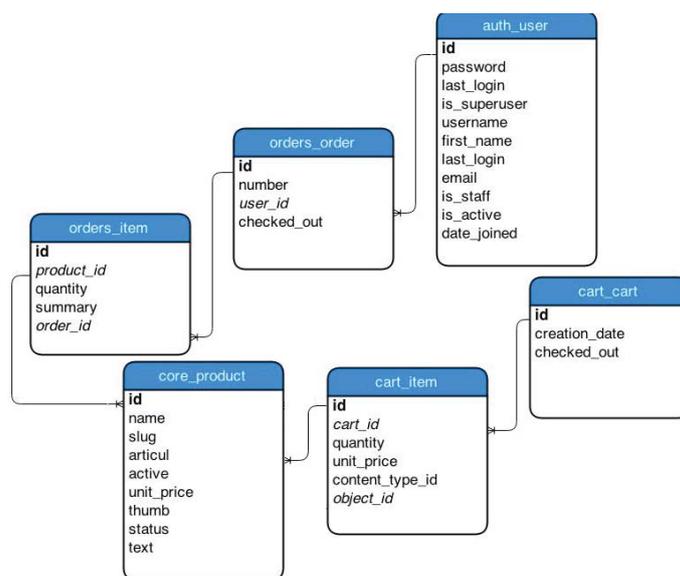


Рис. 1. ER-Модель интерактивной подсистемы подбора сопутствующих товаров

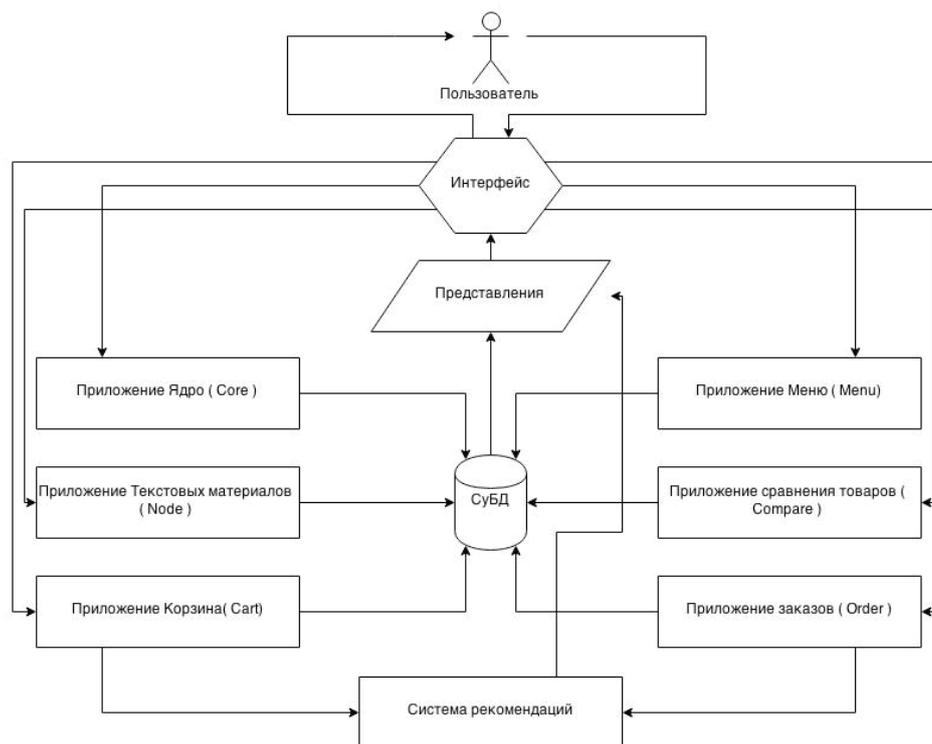


Рис. 2. Взаимодействие модулей

На рис. 2 показана схема взаимодействия модулей:

1. Menu – отвечает за отображение, редактирование, добавление любого типа меню. Состав приложения: models.py с классами типа меню и элемента меню; templatetags с двумя шаблонными тегами для вывода типа меню и его элементов.

2. Core – основное приложение в проекте. На его моделях строится основная структура интернет-магазина. Приложение содержит представления для работы сайта. Состав приложения: models.py с классами для позиции товара, динамичных категорий товара и полей для динамичных категорий; widgets.py – для перезаписи стандартного класса MultiFileInput для загрузки нескольких изображений; templatetags с тремя шаблонными тегами; views.py с 25 представлениями (представление отвечающее за корзину, каталог, главную страницу, карточку товара и т.д.); recommendations.py – пользовательский файл для выборки рекомендуемого товара (входными данными является объект корзины, выходными данными является список рекомендуемого товара). Именно в этом файле реализован алгоритм кластеризации.

3. Goods – приложение с динамическими моделями товаров. Состав приложения: models.py – не содержит классов, т.к. все модели для данного приложения создаются динамически.

4. Node – для обычных текстовых страниц разделяющихся на категории. Состав приложения: models.py с классами категорий и элементов текстового материала и views.py с представлениями для текстовых материалов.

5. Cart – содержит в себе основную логику корзины для интернет-магазина. Состав приложения: models.py с классами корзины и элементов корзины; cart.py – пользовательский файл, включающий в себя создание, удаление и прочее манипулирование как с самим объектом корзины, так и с элементами корзины.

6. Compare – содержит основную логику для сравнения товара. Состав приложения: models.py с классами сравнения и элементов сравнения; compare.py – пользовательский файл, включающий в себя создание удаление и прочее манипулирование как с самим объектом сравнения, так и с элементами сравнения.

7. Orders – содержит подробную информацию о заказе, ее элементах, сумме заказа, а так же о пользователе, который осуществил заказ. Состав приложения заказа: models.py с классами заказов и содержимого заказа.

8. Meta – представляет собой семантическое ядро для продвижения сайта в интернете; создания контекстной рекламы и т.д. Содержимое приложения: models.py содержит один абстрактный класс, для которого не создается таблица в базе данных. Класс наследник содержит 3 дополнительных поля (meta\_title, meta\_keywords и meta\_description). Все классы, которые нуждаются в 3 дополнительных полях для поисковых систем, наследуются от данного класса.

В результате выполнения данного проекта был проведен анализ существующих решений, на основе которого был выбран метод подбора сопутствующих товаров при покупке в интернет-магазине. Выполнена реализация приложений и разработана подсистема, которая была внедрена в опытную эксплуатацию.

#### Список литературы

1. Кольчугина Е.А., Макарь В.А. Метод коллаборативной фильтрации для масштабируемых рекомендательных систем // Современные научные исследования и инновации. – 2012. – № 6. [Электронный ресурс]. – URL: <http://web.snauka.ru/issues/2012/06/14316>
2. Морозов Н.В., Шукенбаев А.Б., Шукенбаева Н.Ш. Подсистема отправки SMS сообщений // Современные наукоемкие технологии. – 2014. – № 5 (2). – С. 212-214.
3. Конькова Ю.В., Шукенбаев А.Б., Шукенбаева Н.Ш. Анализ эффективности сайта и разработка предложений по ее повышению: сборник трудов международной научно-практической конференции «Современные информационные технологии в профессиональной деятельности (СИТ-2014)». – М.: МФЮА, 2014. – С. 111-119.