

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра дискретной математики и информационных технологий

**РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО ВЕБ-ПРИЛОЖЕНИЯ
АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ**

студента 2 курса 271 группы
направления 09.04.01 — Информатика и вычислительная техника
факультета КНиИТ
Курносова Юрия Святославовича

Научный руководитель

д-р экон. наук

Л. В. Кальянов

Заведующий кафедрой

к. ф.-м. н.

Л. Б. Тяпаев

Саратов 2021

ВВЕДЕНИЕ

Впечатление от современных веб-приложений в основном складывается за счет насыщенного возможностями пользовательского интерфейса (user interface, UI). Менее известным аспектом современных приложений является использование методов, которые позволяют осуществлять интеллектуальную обработку информации и добавлять полезные свойства, которые не могут быть обеспечены другими средствами. Многочисленные примеры успешных проектов, использующих такие методики, включают известные бренды, такие как Netflix, Amazon и Google.

Данная магистерская работа посвящена созданию веб-приложения, в котором применяются методики интеллектуальной обработки информации. Для достижения данной цели необходимо:

- ознакомиться с понятием интеллектуального приложения;
- изучить алгоритмы интеллектуальной обработки информации;
- освоить стек технологий, необходимый для реализации веб-приложения;
- применить изученные технологии для разработки новостного агрегатора с элементами интеллектуального приложения.

Магистерская работа состоит из введения, 5 разделов, заключения, списка использованных источников и 7 приложений. Общий объем работы – 68 страниц, из них 51 страница – основное содержание, включая 13 рисунков, цифровой носитель в качестве приложения, список использованных источников информации – 24 наименования.

Научная новизна работы состоит в нахождении и применении алгоритма выработки персональных рекомендаций для конкретной задачи, а именно, рекомендации новостей пользователю по его предпочтениям.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Интеллектуальные приложения» посвящен определению понятия интеллектуального приложения и рассмотрению примеров интеллектуальных приложений.

Интеллектуальные приложения - это приложения, которые учитывают данные, введенные каждым пользователем и его поведение в системе в течение некоторого временного периода, а также другую потенциально полезную информацию, которая может быть доступна. Эти приложения проектируются для автоматического обучения, поэтому они могут понимать вводимые пользователем данные или его поведение, или и то и другое, и соответствующим образом настраивать свой отклик.

Базовые элементы, необходимые для создания интеллектуального приложения:

1. Агрегированный контент – это большой объем данных, который относится к конкретному приложению.
2. Ссылочные структуры – эти структуры обеспечивают одну или несколько структурных и семантических интерпретаций контента.
3. Алгоритмы – этот элемент относится к модульному уровню, который позволяет приложению задействовать информацию, скрытую в данных, и использовать ее в целях предсказания, абстракции и улучшенного взаимодействия с их пользователями.

Примерами интеллектуальных приложений являются:

- Сайты социальных сетей.
- Гибридные приложения (мэшапы).
- Порталы и, в частности, новостные порталы
- Вики-сайты.
- Сайты общего доступа к медиафайлам.
- Онлайн-игры.

Второй раздел «Категории алгоритмов» посвящен обзору алгоритмов интеллектуальной обработки информации.

В данном разделе рассматриваются четыре важных категории алгоритмов:

- Поиск. Под поисковой системой понимают компонент или библиотеку, вообще, программное решение, которое независимо поддерживает свою

собственную базу данных (на самом деле это может быть и СУБД, и просто файлы и распределенная платформа хранения) документов, в которых, собственно, и происходит поиск, предоставляет сторонним приложениям добавлять, удалять и обновлять документы в этой базе.

- Выработка рекомендаций. Рекомендательные системы в основном используются для предложения клиенту в режиме реального времени продуктов (одежды, фильмов, книг), которые вероятно его заинтересуют.
- Кластеризация. Кластеризация (или кластерный анализ) — это задача разбиения множества объектов на группы, называемые кластерами. Внутри каждой группы должны быть «похожие» объекты, а объекты из разных групп должны быть как можно более разными. Основное отличие кластеризации от классификации состоит в том, что список групп четко не определен и определяется в процессе работы алгоритма.
- Классификация. Классификация – это система упорядоченного распределения объектов на какие-либо подразделения на основе определенных признаков (технологических, функциональных и др.).

Третий раздел «Используемый стек технологий» посвящен описанию стека технологий, использованного для разработки веб-приложения.

Для создания веб-приложения использовался следующий стек технологий:

- язык программирования PHP;
- СУБД MySQL;
- фреймворк Symfony;
- Doctrine ORM;
- шаблонизатор Twig;
- поисковая система Elasticsearch;
- язык разметки HTML;
- язык таблиц стилей CSS;
- язык программирования JavaScript;
- фреймворк Bootstrap;
- библиотека JQuery;
- среда разработки PhpStorm.

Четвертый раздел «Разработка веб-приложения» посвящен описанию процесса реализации веб-приложения.

Разработка веб-приложения была разбита на три основные стадии:

- выдвижение требований к веб-приложению;
- разработка серверной части;
- разработка дизайна.

К веб-приложению были выдвинуты следующие требования:

- должно обладать удобным и понятным для пользователя интерфейсом;
- иметь функцию регистрации в системе;
- иметь функцию формирования маски интересов пользователя;
- иметь функцию редактирования маски интересов пользователя;
- иметь функцию предоставления новостей по маске интересов;
- иметь функцию фильтрации новостей по категориям;
- иметь функцию просмотра новости из оригинального источника;
- иметь функцию выработки персональных рекомендаций;
- иметь функцию поиска новостей.

Разработка серверной части веб-приложения была поделена на следующие этапы:

- Проектирование БД. Итогом проектирования стала схема из пяти таблиц: `news_category` (список новостных категорий), `usr` (данные о пользователях), `user_news_category` (информация о предпочтениях пользователя), `news` (список новостей), `user_news` (данные о новостях, понравившихся пользователям).
- Подключение к БД. Подключение к базе данных в Symfony происходит следующим образом: необходимо в конфигурационном файле `.ENV` указать драйвер для работы с БД, адрес хоста, имя БД, имя пользователя и пароль.
- Создание классов сущностей. Хранение и чтение информации из БД в Symfony осуществляется через Doctrine-библиотеку. Doctrine предполагает создание классов сущностей, в которых поля являются отображением полей таблицы БД. А для получения и установки значения этих полей имеются методы `get` и `set`. Типы и свойства полей таблиц БД передаются через аннотации. Они описывают сущность и сообщают ORM Doctrine, каким образом установить соответствие между ней и таблицей базы данных.
- Создание БД и ее схемы. Фреймворк Symfony предоставляет множество

консольных команд через скрипт `bin/console`. Одни из таких команд: `doctrine:database:create` (создает базу данных на основе конфигураций, описанных в файле `.ENV`) и `doctrine:schema:create` (создает схему базы данных на основе Doctrine-сущностей, описанных в папке `src\Entity`). Именно данные консольные команды были использованы для создания базы данных проекта.

- Создание контроллеров. Для реализации функционала приложения были созданы следующие контроллеры: `SecurityController::login` (вход в систему), `SecurityController::logout` (выход из системы), `UserController::new` (создание пользователя), `UserController::show` (отображение страницы профиля пользователя), `UserController::edit` (редактирование пользовательских данных), `UserController::enterCategory` (формирование маски интересов пользователя), `NewsController::add` (добавление новостей из RSS-сервиса Яндекс.Новости в базу данных в таблицу `news`), `NewsController::showOriginal` (просмотр новости из оригинального источника), `NewsController::showUserNews` (отображение новостной ленты пользователя), `NewsController::scoreNews` (проставление отметки "Нравится" на новость), `NewsController::searchNews` (поиск новостей по запросу пользователя).
- Настройка системы безопасности. Система безопасности в Symfony конфигурируется в файле: `config/packages/security.yaml`. В данном конфигурационном файле были прописаны такие параметры безопасности, как: `encoders` (задает способ шифрования пароля), `form_login` (задает маршрут, по которому будет перенаправлен пользователь, для входа в систему), `logout` (задает маршрут, по которому будет перенаправлен пользователь, для выхода из системы), `access_control` (задает маршруты, разрешенные для авторизованных и неавторизованных пользователей).
- Подключение к RSS-сервису Яндекс.Новости. В Яндекс.Новости существуют отдельные RSS-каналы для разнообразных категорий (политика, спорт, музыка и т.д.). Чтобы получать рассылку по конкретной категории, нужно отправить HTTP-запрос по адресу, который имеет следующий вид: `https://news.yandex.ru/{название категории}.rss`. Ответом на запрос является страница с данными в формате XML, которая со-

держит в себе новостной дайджест.

- Разработка рекомендательной системы. Для реализации рекомендательной системы в приложение была внедрена система оценки новостей пользователем, которая была реализована в контроллере `NewsController::scoreNews`. Новость считается понравившейся пользователю если: пользователь поставил на новость отметку "Нравится пользователь перешел на страницу источника новости, для полного прочтения. Система выработки рекомендаций для конкретного пользователя (далее User) работает по следующему принципу:
 1. Поиск пользователей с максимальным совпадением интересов (пользователи с максимальным числом общих новостных категорий с пользователем User).
 2. Поиск пользователей, среди найденных на предыдущем шаге, для которых количество общих понравившихся новостей с пользователем User будет максимальным.
 3. Получение списка понравившихся новостей найденных пользователей и отбор среди них тех, которых еще нет в списке понравившихся новостей пользователя User.
- Внедрение поискового движка Elasticsearch для поиска новостей. Elasticsearch - это отдельная система, которая ведет собственную документо-ориентированную базу данных и по умолчанию слушает localhost на порту 9200. Для работы с поисковой системой была выбрана библиотека Elasticsearch-PHP, которая является официальным клиентом PHP для Elasticsearch. Для взаимодействия с Elasticsearch в проект был добавлен класс `ElasticsearchService`, в котором реализовано два метода: `ElasticsearchService::indexNews` (отправляет данные в формате JSON в поисковую систему для индексации. Данный метод вызывается при добавлении новостей из RSS-сервиса Яндекс.Новости в базу данных, то есть при обращении к контроллеру `NewsController::add`), `ElasticsearchService::searchNewsByPhrase` (поиск новостей по определенной фразе, которую задает пользователь. Поиск в поисковой системе осуществляется по заголовку и содержанию новости).

Разработка дизайна веб-приложения была разбита на следующие эта-

пы:

- Подключение фреймворка Bootstrap и библиотеки JQuery. В Symfony существует базовое представление, которое используется при генерации всех страниц. Данный файл можно найти в каталоге: `templates\base.html.twig`. Именно в файл `base.html.twig` между тегами `<head>` и `</head>` добавляются теги `<script>` с атрибутом `src`, в котором указывается путь к файлам `bootstrap.min.css`, `bootstrap.min.js` и `jquery.min.js`.
- Верстка HTML-страниц. Для взаимодействия пользователя с приложением в проект были добавлены следующие HTML-страницы: `templates/user/new.html.twig` (страница создания пользователя), `templates/user/edit.html.twig` (страница редактирования пользовательских данных), `templates/user/show.html.twig` (страница пользовательского профиля), `templates/user/enter_news.html.twig` (страница формирования пользовательских предпочтений), `templates/news/index.html.twig` (новостная лента), `templates/login.html.twig` (форма для входа в систему).
- применение CSS-стилей фреймворка Bootstrap. Сразу после интеграции фреймворка в проект будет применен шаблон начальной загрузки Bootstrap, который сразу сделает веб-дизайн адаптивным, то есть обеспечит правильное отображение веб-приложения на различных устройствах, подключенных к интернету и динамически подстроится под заданные размеры окна браузера. Применение CSS-стилей в фреймворке происходит через классы, которые прописываются в HTML-теги. Это довольно простой метод, для реализации которого требуются самые базовые знания HTML и CSS.

Пятый раздел «Демонстрация веб-приложения» посвящен демонстрации работы созданного веб-приложения.

Была произведена демонстрация следующих страниц веб-приложения:

- страница авторизации пользователя;
- страница регистрации пользователя;
- страница формирования интересов;
- страница пользовательского профиля;
- главная страница, содержащая в себе новостную ленту, фильтр по новостным категориям и форму для поиска новостей по поисковому запросу пользователя.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной магистерской работы была изучена специальная литература и документация, которые помогли в решении поставленных задач. Было создано веб-приложение, работающее как агрегатор новостей, с элементами интеллектуального приложения. Были решены следующие задачи:

- изучено понятие интеллектуального приложения;
- изучены алгоритмы интеллектуальной обработки информации;
- освоен стек технологий, необходимый для реализации веб-приложения;
- получены навыки использования RSS-сервисов;
- получены навыки реализации рекомендательных систем;
- получены навыки внедрения и использования поисковых движков;
- освоена разработка веб-приложения в среде разработки PhpStorm.

Все задачи решены, поставленная цель магистерской работы выполнена.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Марманис Х., Бабенко Д. Алгоритмы интеллектуального Интернета. — СПб.: «Символ-Плюс», 2011. — С. 23-35.
- 2 Принципы проектирования интеллектуальных веб-приложений [Электронный ресурс] // URL: <https://myslide.ru/presentation/skachat-principy-proektirovaniya-intellektualnyx-vebprilozhenij-lekciya-7-8> (Дата обращения: 02.10.2020). Загл. с экрана. Яз. рус.
- 3 Полнотекстовый поиск в веб-проектах: Sphinx, Apache Lucene, Харіан [Электронный ресурс] // URL: <https://abrdev.com/?p=380> (Дата обращения: 15.10.2020). Загл. с экрана. Яз. рус.
- 4 Lucene.Net Documentation [Электронный ресурс] // URL: <http://lucenenet.apache.org/docs.html> (Дата обращения: 17.10.2020). Загл. с экрана. Яз. англ.
- 5 Системы выработки рекомендаций [Электронный ресурс] // URL: <http://businessdataanalytics.ru/RecommendationSystems.htm> (Дата обращения: 25.10.2020). Загл. с экрана. Яз. рус.
- 6 Гомзин А.Г., Коршунов А.В. Системы рекомендаций: обзор современных подходов. — Москва: «Эксмо», 2012. — С. 401-416.
- 7 Кластеризация [Электронный ресурс] // URL: <https://dic.academic.ru/dic.nsf/ruwiki/969164> (Дата обращения: 10.11.2020). Загл. с экрана. Яз. рус.
- 8 Обзор алгоритмов кластеризации данных [Электронный ресурс] // URL: <https://habr.com/ru/post/101338/> (Дата обращения: 15.11.2020). Загл. с экрана. Яз. рус.
- 9 Классификация: понятие, сущность, цели, методы [Электронный ресурс] // URL: https://studopedia.ru/11_10780_klassifikatsiya-ponyatie-sushchnost-tseli-metodi.html (Дата обращения: 25.11.2020). Загл. с экрана. Яз. рус.
- 10 Методы классификации. Алгоритмы классификации. [Электронный ресурс] // URL: <https://infopedia.su/18xd961.html> (Дата обращения: 26.11.2020). Загл. с экрана. Яз. рус.

- 11 МакГарт. М. PHP 7 для начинающих с пошаговыми инструкциями. — Москва: Издательство «Э», 2018. — С. 12-52.
- 12 PHP Documentation [Электронный ресурс] // URL: <http://php.net/docs.php> (Дата обращения: 05.03.2021). Загл. с экрана. Яз. англ.
- 13 Symfony Documentation [Электронный ресурс] // URL: <https://symfony.com/doc/current/index.html> (Дата обращения: 07.03.2021). Загл. с экрана. Яз. англ.
- 14 Twig Documentation [Электронный ресурс] // URL: <https://twig.symfony.com/doc/2.x/> (Дата обращения: 10.03.2021). Загл. с экрана. Яз. англ.
- 15 Doctrine Documentation [Электронный ресурс] // URL: <https://www.doctrine-project.org/projects/doctrine-orm/en/2.6/index.html> (Дата обращения: 12.03.2021). Загл. с экрана. Яз. англ.
- 16 MySQL Documentation [Электронный ресурс] // URL: <https://dev.mysql.com/doc/> (Дата обращения: 15.03.2021). Загл. с экрана. Яз. англ.
- 17 HTML5 [Электронный ресурс] // URL: <http://htmlbook.ru/html5> (Дата обращения: 12.05.2021). Загл. с экрана. Яз. рус.
- 18 Основы CSS [Электронный ресурс] // URL: <https://webdesign-master.ru/blog/html-css/2017-01-10-css-rukovodstvo.html> (Дата обращения: 15.05.2021). Загл. с экрана. Яз. рус.
- 19 Язык JavaScript [Электронный ресурс] // URL: <https://learn.javascript.ru/js> (Дата обращения: 20.05.2021). Загл. с экрана. Яз. рус.
- 20 JQuery [Электронный ресурс] // URL: <https://jquery.com/> (Дата обращения: 23.05.2021). Загл. с экрана. Яз. рус.
- 21 What is Elasticsearch? [Электронный ресурс] // URL: <https://www.elastic.co/what-is/elasticsearch> (Дата обращения: 17.04.2021). Загл. с экрана. Яз. англ.

- 22 Bootstrap Documentation [Электронный ресурс] // URL: <https://getbootstrap.com/docs/4.1/> (Дата обращения: 15.03.2021). Загл. с экрана. Яз. рус.
- 23 Архитектура MVC [Электронный ресурс] // URL: <https://github.com/codedokode/pasta/blob/master/arch/mvc.md> (Дата обращения 08.05.2021). Загл. с экрана. Яз. рус.
- 24 PhpStorm Documentation [Электронный ресурс] // URL: <https://www.jetbrains.com/phpstorm/documentation/> (Дата обращения 16.05.2021) Загл. с экрана. Яз. англ.