

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Новосибирский национальный исследовательский государственный университет»
(Новосибирский государственный университет, НГУ)
Структурное подразделение Новосибирского государственного университета
Высший колледж информатики Университета (ВКИ НГУ)
КАФЕДРА ИНФОРМАТИКИ

**РАЗРАБОТКА ЧАТ-БОТА «КНИЖНЫЙ БОТ РЕКОМЕНДАЦИИ
КНИГ» ДЛЯ МЕССЕНДЖЕРА ТЕЛЕГРАМ**
Квалификация программист

Руководитель
директор компании e2e4gu.ru

Потапенко В. А.
« ____ » _____ 2018 г.

Студент 4 курса
гр. 403с

Дугуржап Х. Ш.
« ____ » _____ 2018 г.

Новосибирск
2018

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ	4
ВВЕДЕНИЕ.....	5
1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	6
2 ПОСТАНОВКА ЗАДАЧИ.....	7
3 АНАЛОГИ	8
3.1 Интернет-портал «GoodReads»	8
3.2 Рекомендательный сервис «Имхонет»	8
4 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	9
4.1 Варианты использования	9
4.1.1 Сбор данных о книгах	9
4.1.2 Сбор рецензий о книгах.....	10
4.1.3 Рекомендация книг пользователю	10
4.1.4 Отображение прочитанных книг пользователя	11
4.1.5 Отображение понравившихся книг пользователя	12
4.1.6 Отображение случайной книги	13
4.1.7 Поиск книг по названию.....	14
4.1.8 Отображение руководства пользователя	15
5 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ	16
5.1 Требования к программному обеспечению.....	16
5.2 Требования к аппаратному обеспечению.....	16
5.3 Требования к надежности	16
5.4 Требования к безопасности.....	17
6 ХАРАКТЕРИСТИКА ВЫБРАННЫХ ПРОГРАММНЫХ СРЕД И СРЕДСТВ... ..	18
6.1 Описание программных сред и средств	18
7 СТРУКТУРА ДАННЫХ	21
8 СХЕМА ФУНКЦИОНИРОВАНИЯ ПРОГРАММНОГО СРЕДСТВА.....	26
8.1 Описание классов.....	26
9 ОПТИМИЗАЦИЯ ПРОГРАММНОГО КОДА ПРИЛОЖЕНИЯ	30

10 ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА	31
11 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	33
11.1 Введение.....	33
11.2 Инструкция.....	34
ЗАКЛЮЧЕНИЕ.....	42
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	43
ПРИЛОЖЕНИЕ А.....	44

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

— API (Application Programming Interface) – интерфейс программирования для создания приложений.

— ISBN (International Standard Book Number) – международный стандартный книжный номер.

— ORM (Object-Relational Mapping) – технология, которая связывает различные базы данных с концепциями объектно-ориентированных языков программирования.

— SSL (Secure Sockets Layer) – криптографический протокол, который обеспечивает шифрованное соединение между клиентом и сервером. Информация, которая передается между клиент-сервером, защищена от третьих лиц.

— XSS (Cross-Site Scripting) – тип атаки на веб-системы, заключающийся во внедрении в выдаваемую веб-системой страницу вредоносного кода (который будет выполнен на компьютере пользователя при открытии им этой страницы) и взаимодействии этого кода с веб-сервером злоумышленника.

— Бот – специальная программа, которая выполняет автоматически и по заданному расписанию какие-либо действия через интерфейсы, предназначенные для людей.

— БД – база данных.

— Мессенджер – программа по обмену мгновенными сообщениями.

— РС – рекомендательная система.

— СУБД – система управления базами данных.

— Телеграм-бот – сторонние приложения, которые работают внутри мессенджера Телеграм.

— Чат-бот – программа, работающая внутри мессенджера.

ВВЕДЕНИЕ

Темой данной выпускной квалификационной работы является разработка чат-бота «Книжный бот рекомендации книг» для мессенджера Телеграм.

Целью данной темы является разработка чат-бота для мессенджера Телеграм, который рекомендует книги на основе собранной информации о книгах и отзывах о них с нескольких сайтов.

Для достижения цели были поставлены следующие задачи:

- Исследовать предметную область.
- Разработать функциональные и нефункциональные требования.
- Спроектировать структуру базы данных и выбрать архитектуру приложения.
- Разработать модули для сбора данных, рекомендательную систему и пользовательский интерфейс в виде чат-бота.

На основе вышеперечисленных задач были сформулированы следующие функциональные требования для чат-бота:

- Просмотр рекомендаций.
- Просмотр книг, добавленных в прочитанные или понравившиеся.
- Поиск книг по авторам, названиям, жанрам.
- Просмотр справки.

Данный бот рассчитан на аудиторию пользователей, увлекающихся художественной литературой. Рекомендации позволят лучше ориентироваться в популярной литературе, сократить время на поиск книги.

Используемые программные средства и технологии:

- Язык программирования PHP.
- База данных MySQL.
- Платформа Node.JS.
- Node.JS Telegram Bot API.

1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Телеграм – это кроссплатформенный мессенджер. Мессенджер позволяет обмениваться сообщениями и медиа-файлами разных форматов. Мессенджер позволяет разрабатывать боты для осуществления поиска в интернете или выполнения иных задач. Боты в Телеграм делятся на несколько направлений:

- Чат-боты – вид ботов, имитирующий общение на заданную пользователем тематику.
- Боты-информаторы – отдельный вид ботов, главная цель которых – информирование пользователя о тех или иных событиях (новости, мероприятия, публикации и т.п.).
- Игровые боты – боты, в которых можно поиграть в различные игры.
- Боты-ассистенты – боты, разработанные различными онлайн-сервисами как дополнение к основной веб-версии [1].

Рекомендательные системы – программы, которые пытаются предсказать, что хотят видеть те или иные пользователи.

Рекомендации вырабатываются для каждого пользователя отдельно на основе его предыдущего действия или прошлой активности на конкретном веб-ресурсе или на основе поведения предыдущих участников процесса.

Виды рекомендательных систем, определяемых по подходам, которые они применяют:

- Коллаборативная фильтрация – рекомендации с использованием данной методики выдаются, основываясь на поведенческих характеристиках одного человека или группы людей.
- Контентная фильтрация – рекомендации формируются исходя из поведения человека [2].

2 ПОСТАНОВКА ЗАДАЧИ

Цель работы – проектирование и разработка Телеграм-бота, который дает рекомендации книг на основе собранных данных о рецензиях различных книг с различных сайтов. Бот также включает в себя следующие возможности:

- Поиск книг по названиям.
- Составление списка прочитанных и понравившихся книг.

Рекомендательная система собирает рецензии людей на различные книги с сайтов LiveLib [3] и ЛитРес [4], и на основе этих рецензий рекомендует книги пользователям бота. Собранные данные включает в себя все рецензии и книги пользователя книжного сайта. Рекомендация книг зависит от предпочтений пользователя, например, если пользователю нравится рекомендуемая книга или читал её, то система будет рекомендовать любимые книги рецензентов, которые оставили положительный отзыв понравившейся или прочитанной книге пользователя. Другими словами, рекомендательная система работает по принципу, что люди, которым нравится одна и та же книга, могут иметь схожие интересы. Исходя из этого, рекомендует книги пользователям бота.

Исходя из всего вышеперечисленного, была поставлена задача научным руководителем Потапенко Вадимом Александровичем – разработать модуль для сбора данных о книгах и отзывах, рекомендательную систему и реализовать её интерфейс в виде Телеграм-бота. Рекомендательная система предназначена для того, чтобы рекомендовать книги пользователям, а также для осуществления поиска книг по названиям и для сохранения любимых и прочитанных книг в закладки. Название, автор, ISBN, обложка и прочая информация сохраненных книг хранится в БД.

3 АНАЛОГИ

В сети Интернет существует огромное количество рекомендательных систем разной тематики. В качестве аналогов можно рассматривать два сайта, которые имеют рекомендательную систему книг – это интернет-портал «GoodReads» и рекомендательный сервис «Имхонет».

3.1 Интернет-портал «GoodReads»

GoodReads [5] – интернет-портал, социальная сеть для любителей книг. Отбор необходимой рекомендуемой литературы происходит по заданным знаниям о том, какие книги нравятся читателю, каким книгам отдает предпочтение, какие книги нравятся людям со схожими интересами.

Положительные стороны портала. Портал предоставляет возможность свободного доступа к обширной базе данных книг, а также пользователи могут добавлять свои любимые книги в свои личные полки.

Отрицательные стороны портала. Пользователям необходимо зарегистрироваться, чтобы получить рекомендации книг и отметить не менее 20 книг как предпочитаемые, а также данный сервис является больше социальной сетью, чем рекомендательным сервисом.

3.2 Рекомендательный сервис «Имхонет»

Имхонет [6] – интеллектуальная среда развлечений. Сервис рекомендует не только книги, но и фильмы, и музыку. Для выработки персональных рекомендаций собираются различные пользовательские сигналы. На основе собранных сигналов набором алгоритмов машинного обучения и анализа данных составляется его вкусовой профиль. Сервис сравнивает профили разных пользователей и формирует сообщества единомышленников – людей с близкими вкусами. Эти группы единомышленников используются для формирования прогнозных оценок для элементов разных типов.

4 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

В чат-боте предусмотрена только одна категория пользователей, которая имеет возможность выполнять следующие функции:

- Просмотр рекомендаций.
- Просмотр книг, добавленных в прочитанные или понравившиеся.
- Поиск книг по названиям.
- Просмотр справки.

В разделах 4.1.1-4.1.2 описаны варианты использования поискового робота для сбора данных о книгах и отзывах. В разделах 4.1.3-4.1.8 описаны все варианты использования бота и его реакцию на совершенные действия пользователя. В вариантах использования имеются следующие роли, которые выполняют те или иные действия:

- Пользователь – человек, который пользуется ботом.
- РС – рекомендательная система.
- Бот – чат-бот, реализующий интерфейс рекомендательной системы.
- Поисковый робот – модуль, который собирает данные о книгах и рецензиях.

4.1 Варианты использования

4.1.1 Сбор данных о книгах

Целью этого варианта использования является сбор данных о книгах с различных сайтов. Для достижения этой цели поисковому роботу нужно пройти следующие этапы:

- Класс для сбора информации книг передает запрос в БД о том, по какой ссылке собирать данные о книге.
- БД передает ссылку книги для сбора данных.
- Класс проверяет по ISBN книги, существует ли эта книга в БД. Если нет, переходит в следующий этап.

- Класс делает парсинг страницы книги. Собирает из страницы необходимую информацию книги, а именно название, описание, авторов, обложку, рейтинг и жанры.
- Класс передает собранные данные о книге в БД для хранения.

4.1.2 Сбор рецензий о книгах

Целью этого варианта использования является сбор рецензий о книгах с различных сайтов. Для достижения этой цели поисковому роботу нужно пройти следующие этапы:

- Класс для сбора рецензий книг передает запрос в БД о том, рецензии какой книги собирать.
- БД передает ссылку рецензий книг для их сбора.
- Класс делает парсинг страницы рецензий книги. Собирает необходимую информацию о рецензиях из страницы рецензий, а именно, имя рецензента и поставленную оценку.
- Класс передает собранные данные в БД для хранения.
- В базу данных отдельно хранится информация о рецензиях и о рецензентах.

4.1.3 Рекомендация книг пользователю

Целью этого варианта использования является рекомендация книг пользователю на основе собранных данных о рецензиях. Для достижения этой цели нужно выполнить следующие этапы:

- Бот передает уникальный номер пользователя РС.
- РС ищет, какие книги были понравившимися пользователю. Если были найдены понравившиеся книги пользователя, переходит в следующий этап, иначе выберет любую книгу.
- РС ищет всех рецензентов, которые написали рецензии этим книгам.
- РС выбирает одного рецензента, который оставил высокую оценку одной из этих книг.

- РС ищет, к каким книгам написал рецензии тот рецензент и оставлял высокие оценки.
- Система выбирает одну книгу рецензента, который он высоко оценил (5 или 4).
- Система отправляет эту книгу боту для отображения пользователю.
- Бот показывает рекомендованную книгу.

На рисунке 1 показана UML-диаграмма варианта использования «Рекомендация книг пользователю».

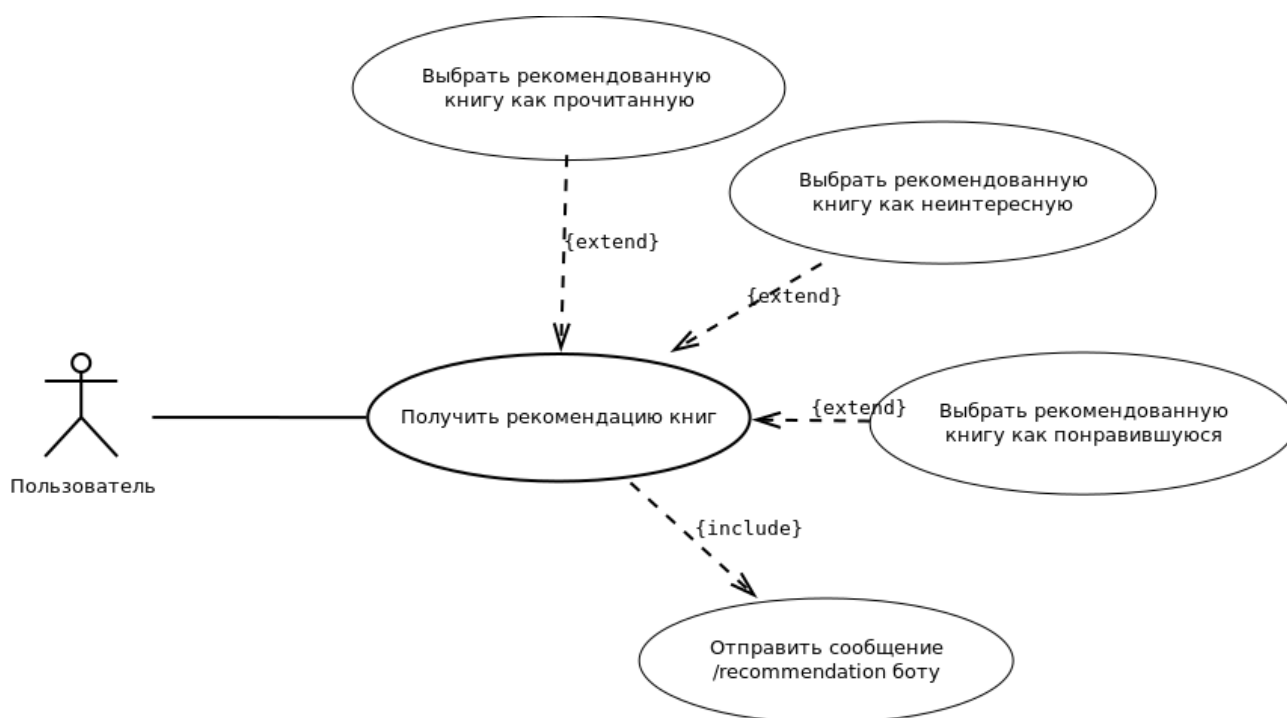


Рисунок 1

4.1.4 Отображение прочитанных книг пользователя

Целью этого варианта использования является отображение прочитанных книг пользователя. Для достижения этой цели нужно выполнить следующие этапы:

- Пользователь отправляет боту сообщение «/read».
- Бот отправляет запрос в БД для получения прочитанных книг пользователя.

- БД возвращает все прочитанные книги пользователя. Если нет прочитанных книг, возвращает сообщение о том, что прочитанных книг не найден.
- Бот отображает информацию одной прочитанной книги пользователя и соответствующие кнопки («предыдущая» и «следующая») для отображения других прочитанных книг. Если нет прочитанных книг, выводится сообщение о том, что книги не найдены.

На рисунке 2 показана UML-диаграмма варианта использования «Отображение прочитанных книг пользователя».

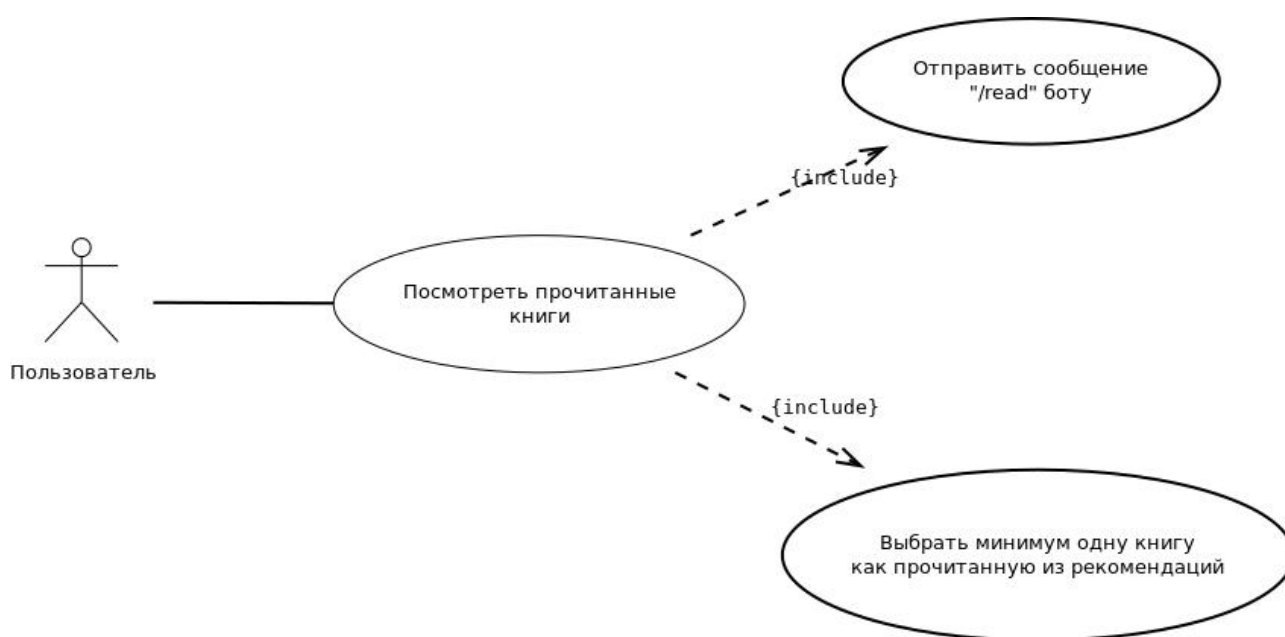


Рисунок 2

4.1.5 Отображение понравившихся книг пользователя

Целью этого варианта использования является отображение понравившихся книг пользователя. Для достижения этой цели нужно выполнить следующие этапы:

- Пользователь отправляет боту сообщение «/like».
- Бот отправляет запрос в БД для получения понравившихся книг пользователя.
- БД возвращает все понравившиеся книги пользователя. Если нет понравившихся книг, возвращает сообщение о том, что понравившихся книг не найден.

— Бот отображает информацию одной понравившейся книги пользователя и соответствующие кнопки («предыдущая» и «следующая») для отображения других понравившихся книг. Если нет понравившихся книг, выводится сообщение о том, что книги не найдены.

На рисунке 3 показана UML-диаграмма варианта использования «Отображение понравившихся книг пользователя».

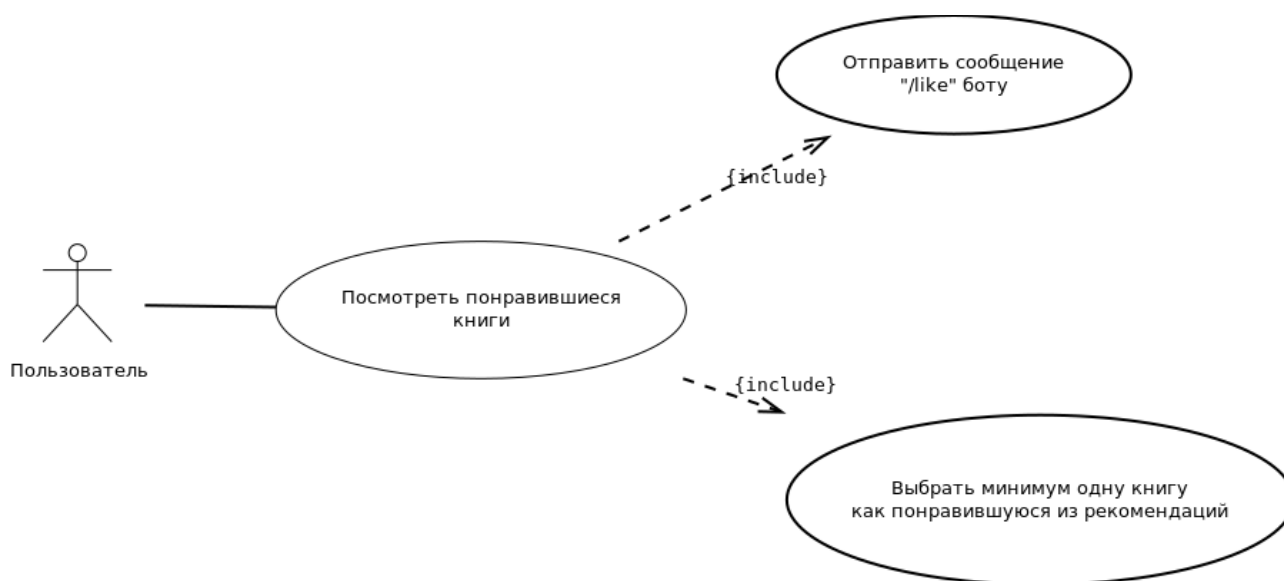


Рисунок 3

4.1.6 Отображение случайной книги

Целью этого варианта использования является отображение случайной книги пользователю. Для достижения этой цели нужно выполнить следующие этапы:

- Пользователь отправляет боту сообщение «/random».
- Бот выбирает одну случайную книгу из БД и отправляет информацию о книге пользователю.
- Бот показывает две кнопки – «Показать другую книгу» и «Добавить в прочитанные книги».

На рисунке 4 показана UML-диаграмма варианта использования «Отображение случайной книги».

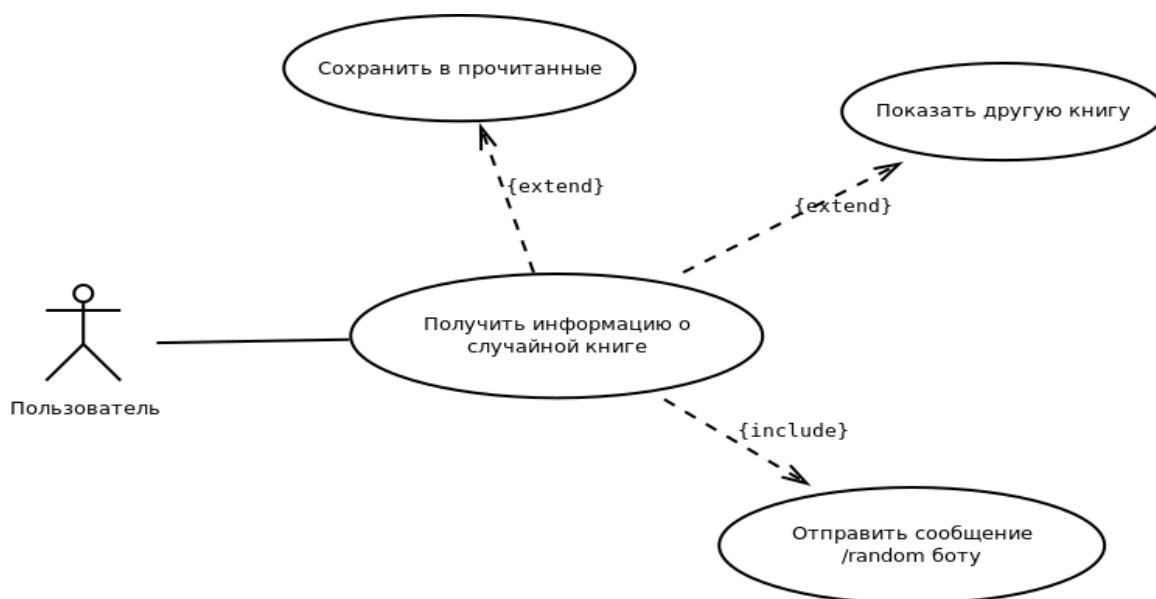


Рисунок 4

4.1.7 Поиск книг по названию

Целью этого варианта использования является поиск книг и отображение их пользователю. Для достижения этой цели нужно выполнить следующие этапы:

- Пользователь отправляет боту сообщение «/search название книги».
- Бот выдает список всех книг, где в названии «название книги».

На рисунке 5 показана UML-диаграмма варианта использования «Поиск книг по названию».



Рисунок 5

4.1.8 Отображение руководства пользователя

Целью этого варианта использования является отображение руководства пользователя. Для достижения этой цели нужно выполнить следующие этапы:

- Пользователь отправляет боту сообщение «/help».
- Бот отображает информацию о боте, все доступные команды бота и краткое описание этих команд.

На рисунке 6 показана UML-диаграмма варианта использования «Отображение руководства пользователя».



Рисунок 6

5 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

5.1 Требования к программному обеспечению

Для обеспечения функционирования программного средства необходимо использовать следующие программные средства:

- Базу данных MySQL.
- Интерпретатор PHP.
- Сценарный язык программирования JavaScript.
- Фреймворк CakePHP.
- Программную платформу Node.js.
- Node.JS Телеграм Bot API.
- Sequelize ORM.
- Мессенджер Телеграм.

5.2 Требования к аппаратному обеспечению

Для обеспечения функционирования бота необходимо использовать:

- Количество свободного места на физическом накопителе не меньше 16 ГБ.
- Частота процессоров не ниже 1.5 GHz.

Минимальные требования к устройстве клиента:

- Установленный мессенджер Телеграм.

5.3 Требования к надежности

К боту предъявляются следующие требования в плане надежности:

- Система должна проверять вводимые команды пользователя, если команда корректная, то бот должен показать соответствующий вывод команды, иначе сообщение о том, что такая команда не существует.

5.4 Требования к безопасности

Для обеспечения безопасности бота необходимы следующие требования:

- Бот должен храниться в сервере с SSL-сертификатом для обеспечения шифрования.

6 ХАРАКТЕРИСТИКА ВЫБРАННЫХ ПРОГРАММНЫХ СРЕД И СРЕДСТВ

В таблице 1 представлен тип и название используемых программных сред и средств:

Тип	Название
Операционная система	Ubuntu 16.04 LTS
Среды разработки	Atom
Редактор диаграмм	Dia
Инструмент проектирования БД	MySQL Workbench
Система контроля версий	Git
Языки программирования	PHP, JavaScript
СУБД	MySQL
Программные платформы (фреймворки)	CakePHP, Node.js
Веб-сервис	Bitbucket.org
Другие технологии	Sequelize ORM, GNU Bourne-Again Shell (bash)

Таблица 1

6.1 Описание программных сред и средств

Ubuntu – операционная система, основанная на Debian GNU/Linux.

Atom – графический текстовый редактор. Был выбран для написания модулей чат-бота и системы рекомендаций, учитывая его следующие преимущества: кроссплатформенность, интеграция с системой управления git, подсветка синтаксиса разных языков, открытый исходный код, свободное ПО, огромное количество плагинов, высокая производительность [7].

Dia – свободный кроссплатформенный редактор диаграмм. Он может быть использован для создания различных видов диаграмм: блок-схем алгоритмов программ, древовидных схем, статических структур UML, баз данных и других.

Основные преимущества: кроссплатформенность, бесплатность [8]. Для создания UML-диаграмм выбор был остановлен на этом редакторе, учитывая его вышеуказанные преимущества.

MySQL Workbench – инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL [9].

Git – распределенная система контроля версий.

PHP – сценарный язык программирования общего назначения, применяемый в основном для разработки веб-приложений.

JavaScript – мультипарадигменный сценарный язык программирования.

MySQL – свободная реляционная система управления базами данных.

CakePHP – фреймворк для создания веб-приложений, написанный на языке PHP. Преимущества: открытый исходный код (лицензия MIT), поддержка множества плагинов, поддержка множества СУБД, имеется автогенератор кода bake, консольная интеграция, миграции и другие. Данный фреймворк был выбран для разработки модуля для сбора данных, так как он предоставляет API для написания консольных приложений.

Node.js – программная платформа, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js использует событийно-ориентированный и неблокирующий модель ввода-вывода, что делает его легким и эффективным, а также добавляет возможность JavaScript подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода [10]. Платформа была выбрана для разработки рекомендательной системы и чат-бота, так как она удобна для работы с большими данными и предоставляет огромное количество библиотек с открытым исходным кодом.

Node.js Telegram Bot API – библиотека для разработки Телеграм-бота на программной платформе Node.js. Преимущества: открытый исходный код

(лицензия MIT), имеется документация. Данная библиотека была выбрана для разработки модуля чат-бота, учитывая её вышеуказанные преимущества.

Bitbucket.org – веб-сервис для репозиторий и их совместной разработки, основанный на системе контроля версий Git и Mercurial. Веб-сервис был выбран для хранения исходного кода программного продукта, так как он предоставляет возможность хранить проекты бесплатно и приватно.

Sequelize ORM – библиотека для подключения к БД, ORM на основе Promise, разработанный для программной платформы Node.js v4 и выше. Sequelize поддерживает диалекты PostgreSQL, MySQL, SQLite, MSSQL. Данная библиотека была выбрана, так как он предоставляет удобный API для написания запросов к БД.

GNU Bourne-Again Shell (bash) – усовершенствованная командная оболочка. Является стандартным интерпретатором командной строки UNIX подобных систем.

7 СТРУКТУРА ДАННЫХ

Для реализации рекомендательной системы и чат-бота была разработана следующая структура данных. Общая схема данных представлена на рисунке 7.

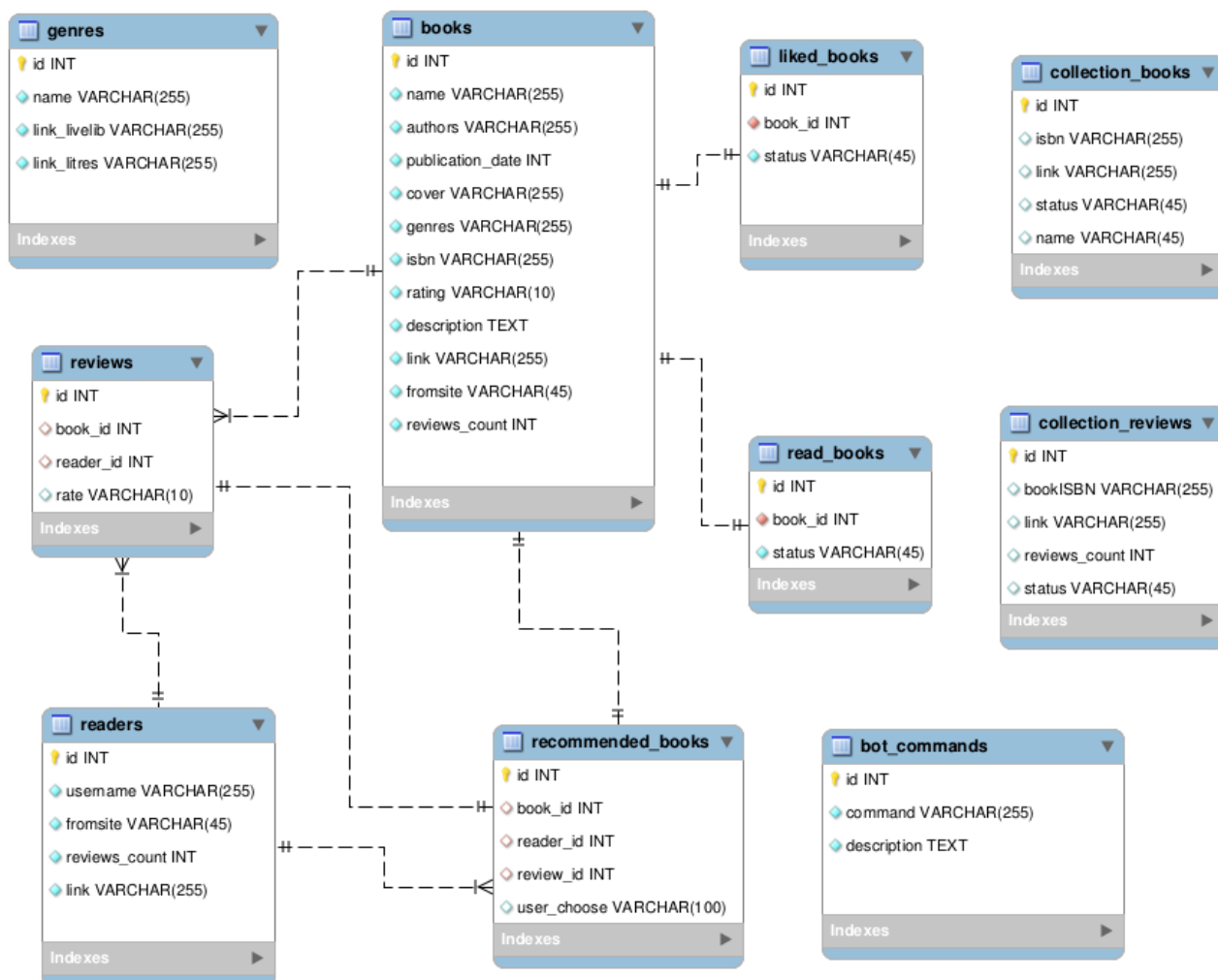


Рисунок 7

В таблице Books хранится информация о книгах. На рисунке 8 представлено описание структуры таблицы Books.

Имя атрибута	Описание атрибута	Тип атрибута
id	ID книги	Счетчик
name	Название книги	Текстовый
authors	Авторы книги	Текстовый
publication_date	Дата публикации книги	Числовой
cover	Обложка книги	Текстовый
genres	Жанры книги	Текстовый
isbn	ISBN книги	Текстовый
rating	Общий рейтинг книги на сайте livelib	Текстовый
description	Описание книги	Текстовый
link	Ссылка книги на сайте livelib или litres	Текстовый
fromsite	С какого сайта была собрана информация о книге (livelib или litres)	Текстовый
reviews_count	Количество рецензий книги	Числовой
quote_id	ID цитаты из книги	Числовой

Рисунок 8

В таблице Bot_commands хранится информация о командах, выполняемых ботом. На рисунке 9 представлено описание структуры таблицы Bot_commands.

Название атрибута	Описание атрибута	Тип атрибута
id	ID команды	Счетчик
command	Название команды	Текстовый
description	Описание команды	Текстовый

Рисунок 9

В таблице Genres хранится название и ссылки жанров. На рисунке 10 представлено описание структуры таблицы Genres.

Название атрибута	Описание атрибута	Тип атрибута
id	ID жанра	Счетчик
name	Полное название жанра	Текстовый
link_livelib	Ссылка жанра в livelib	Текстовый
link_litres	Ссылка жанра в litres	Текстовый

Рисунок 10

В таблицах Liked_books и read_books хранится информация о книгах, понравившихся (прочитанных) пользователям бота. На рисунке 11 представлено описание структуры таблицы Liked_books.

Название атрибута	Описание атрибута	Тип атрибута
id	ID записи	Счетчик
user_id	ID пользователя, который сохранил книгу	Числовой
book_id	ID понравившейся (прочитанной) книги	Числовой
status	Выбор пользователя (добавлено, удалено)	Текстовый

Рисунок 11

В таблице Readers хранится информация о читателях, пишущих рецензии на сайтах. На рисунке 12 представлено описание структуры таблицы Readers.

Название атрибута	Описание атрибута	Тип атрибута
id	ID читателя	Счетчик
username	Имя пользователя	Текстовый
fromsite	С какого сайта читатель	Текстовый
reviews_count	Количество написанных рецензий читателя	Числовой
link	Ссылка на профиль читателя	Текстовый

Рисунок 12

В таблице Recommended_books хранится информация о книгах, которые нужно рекомендовать пользователям бота. На рисунке 13 представлено описание структуры таблицы Recommended_books.

Название атрибута	Описание атрибута	Тип атрибута
id	ID рекомендации	Счетчик
book_id	ID книги	Числовой
reader_id	ID рецензента	Числовой
review_id	ID рецензии	Числовой
user_id	ID пользователя	Числовой
user_choose	Выбор пользователя (нравится, читал, не нравится)	Текстовый

Рисунок 13

В таблице Reviews хранится информация о рецензиях. На рисунке 14 представлено описание структуры таблицы Reviews.

Название атрибута	Описание атрибута	Тип атрибута
id	ID рецензии	Счетчик
book_id	ID книги	Числовой
reader_id	ID рецензента	Числовой
rate	Проставленная оценка рецензента книге	Числовой

Рисунок 14

В таблице Collection_books хранится название, ISBN, ссылки собираемых книг. На рисунке 15 представлено описание структуры таблицы Collection_books.

Название атрибута	Описание атрибута	Тип атрибута
id	ID записи	Счетчик
isbn	ISBN книги	Текстовый
link	Ссылка книги	Текстовый
status	Статус сбора	Текстовый
name	Название книги	Текстовый

Рисунок 15

В таблице Collection_reviews хранится ISBN, ссылки рецензий собираемых рецензий. На рисунке 16 представлено описание структуры таблицы Collection_reviews.

Название атрибута	Описание атрибута	Тип атрибута
id	ID записи	Счетчик
isbn	ISBN книги	Текстовый
link	Ссылка на отзывы книги	Текстовый
reviews_count	Количество рецензий	Числовой
status	Статус сбора	Текстовый

Рисунок 16

8 СХЕМА ФУНКЦИОНИРОВАНИЯ ПРОГРАММНОГО СРЕДСТВА

Разработка программного продукта делится на 3 части:

- Разработка модуля для сбора данных о книгах и рецензиях.
- Разработка механизма рекомендации книг.
- Разработка пользовательского интерфейса в виде Телеграм-бота.

8.1 Описание классов

Модуль для сбора данных о книгах и отзывах состоит из двух классов – это классы «CollectionBooksShell» и «CollectionReviewsShell». Также существует классы, реализующие модели всех сущностей. Ниже приведены описание всех используемых классов.

На рисунке 17 показаны диаграммы всех используемых классов.

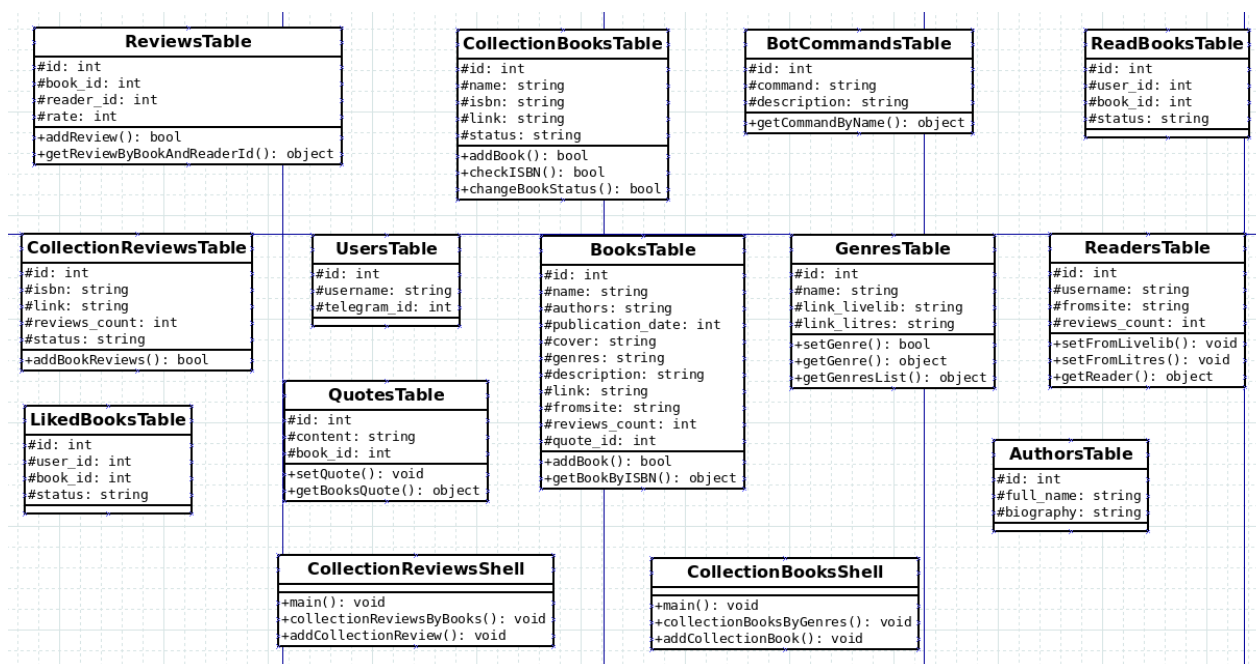


Рисунок 17

Класс «CollectionBooksShell» – этот класс реализует сбор данных о книгах из сайтов livelib.ru и litres.ru и автоматизирует данный процесс. Листинг программного кода представлен в приложении А. У данного класса есть три метода:

- Метод «collectionBooksByGenres» – метод собирает ссылку, название и isbn всех книг определенного жанра и передает собранные данные классу «CollectionBooksTable» для дальнейшего занесения полученных данных в таблицу «collection_books» в БД.
- Метод «addBook» – метод собирает подробную информацию о книгах, добавленных в таблицу «collection_books», затем, если собранные данные корректны, передает их классу «BooksTable» для занесения в таблицу «books» в БД.
- Метод «main» – этот метод вызывает методы выше в назначенное время, таким образом запускает процесс сбора данных о книгах.

Класс «CollectionReviewsShell» – этот класс реализует сбор данных о рецензиях из сайтов «livelib.ru» и «litres.ru» и автоматизирует данный процесс. У него есть три метода:

- Метод «collectionReviewsByBooks» – метод собирает ссылки рецензий определенных книг и передает собранные данные классу «CollectionReviewsTable» для занесения полученных данных в таблицу «collection_reviews» в БД.
- Метод «addCollectionReview» – метод собирает все данные о рецензиях книг, добавленных в таблицу «collection_reviews», затем, если собранные данные отсутствуют в БД и они корректны, вызывает метод «addReview» класса «ReviewsTable» для добавления рецензии в БД.
- Метод «main» – этот метод вызывает методы выше в назначенное время, таким образом запускает процесс сбора данных о книгах.

Класс «CollectionBooksTable» – класс для получения, обработки, проверки, занесения и передачи данных таблицы «collection_books». Класс является частью модели «collection_books». У данного класса есть три метода:

- Метод «addBook» – добавление ссылки, ISBN, название новых книг в БД. У него есть входной параметр (массив данных о книге) и

выходной параметр («true/false» в зависимости от того, добавлена книга в БД или нет).

- Метод «checkISBN» – метод проверяет наличие книги в БД по ISBN. Возвращает «true», если была найдена книга, иначе «false».
- Метод «changeBookStatus» – метод меняет статус книги на «added», если были, собрана информация о книге и была добавлена в таблицу «books».

Класс «CollectionReviewsTable» – класс для получения, обработки, проверки, занесения и передачи данных таблицы «collection_reviews». Класс является частью модели «collection_books». У данного класса есть один метод:

- Метод «addBookReviews» – добавление ссылок на рецензии, ISBN, название книг в БД. У него есть входной параметр (массив данных о книге) и выходной параметр («true/false» в зависимости от того, добавлена информация в БД или нет).

Класс «BooksTable» – класс для получения, обработки, проверки, занесения и передачи данных таблицы «books». У данного класса есть два метода:

- Метод «addBook» – метод добавляет собранную информацию о книге в БД. У него есть входной параметр (массив данных книг) и выходной параметр (true/false в зависимости от того, добавлена книга в БД или нет).
- Метод «getBookByISBN» – метод получает данные определенной книги по ISBN. У него есть входной параметр (ISBN книги) и выходной параметр (данные полученной книги из БД).

Класс «ReviewsTable» – класс для получения, обработки, проверки, занесения и передачи данных таблицы «reviews». У данного класса есть два метода:

- Метод «addReview» – метод добавляет собранную информацию о рецензиях в БД. У него есть входной параметр (массив данных о рецензиях) и выходной параметр («true/false» в зависимости от того, добавлена рецензия в БД или нет).

— Метод «getReviewByBookAndReaderId» – метод получает данные рецензии из БД по индивидуальным номерам книг и рецензента. У него есть два входных параметра (индивидуальный номер книги и индивидуальный номер рецензента) и выходной параметр (данные полученной рецензии из БД).

9 ОПТИМИЗАЦИЯ ПРОГРАММНОГО КОДА ПРИЛОЖЕНИЯ

Были проведены анализ и оптимизация программного кода для повышения производительности и эффективности работы рекомендательной системы бота. Было оптимизировано следующее:

- Механизм рекомендательной системы реализован на платформе Node.js. Данная платформа позволяет эффективно обрабатывать операции ввода-вывода с большими данными.
- Запросы получения необходимой информации из БД выполнены с помощью ORM Sequelize для повышения производительности.
- Некоторые участки кода, которые снижают эффективность, были полностью переписаны.
- Для увеличения скорости работы СУБД было минимизировано количество операций ввода-вывода.

10 ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

Отладка – этап разработки чат-бота, на котором выявляют и устраняют ошибки. На этот этап входит такие процессы, как тестирование, поиск ошибок и устранение ошибок.

Тестирование – это процесс выполнения отдельных модулей и команд чат-бота на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения чат-бота. Другими словами, тестирование – это проверка корректной работы функциональностей чат-бота, а также удобства использования. Были разработаны тестовые сценарии для проверки работы функций и команд бота.

В таблице 2 представлен результат тестирования команд бота, другими словами, корректность отображения результата на определенное сообщение пользователя. В первом столбце перечислены сообщения, которые могут быть отправлены пользователем. Во второй столбце ответ на сообщения, иначе говоря, ожидаемый результат пользователя. В третьем столбце результат тестирования.

Действие пользователя	Ожидаемый результат	Результат теста
Нажать на кнопку «start»	На экране появится приветственное сообщение	Пройден
Отправить сообщение «/help»	На экране появится руководство по использованию	Пройден
Отправить сообщение «справка»	На экране появится руководство по использованию	Пройден
Отправить сообщение «/random»	На экране появится информация случайной книги, выбранная системой	Пройден
Отправить сообщение «/recommendation»	На экране появится информация рекомендуемой книги системы и три кнопки «нравится», «читал (а)» и «не нравится»	Пройден
Отправить сообщение «рекомендация»	На экране появится информация рекомендуемой книги системы и три кнопки «нравится», «читал (а)» и «не нравится»	Пройден
Отправить сообщение «/read»	На экране появится список названий всех прочитанных книг пользователя	Пройден
Отправить сообщение «/like»	На экране появится список названий всех книг пользователя, которые были выбраны как «нравится» рекомендаций бота	Пройден
Отправить любое другое сообщение	На экране появится сообщение «К сожалению, бот Вас не понимает. Чтобы просмотреть, какие команды доступны боту, отправьте сообщение справка»	Пройден

Таблица 2

11 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

11.1 Введение

Данное руководство описывает весь функционал чат-бота «Книжный бот рекомендации книг». Оно посвящено для конечного пользователя данного продукта, а именно, для пользователей мессенджера Телеграм, интересующихся художественной литературой. Чат-ботом могут пользоваться все пользователи мессенджера Телеграм.

Основное назначение чат-бота – рекомендация книг. Пользователи будут получать рекомендации бота в двух случаях: пользователи отправляют сообщение боту и подписка на уведомления. Кроме того, пользователи могут осуществить поиск книг и посмотреть сохраненные книги.

Предусмотрены следующие минимальные требования пользования чат-ботом:

- Любое из перечисленных устройств: iPhone, iPad, Android Phone, Windows Phone или персональный компьютер.
- Установленный мессенджер Телеграм на устройстве пользователя или веб-браузер.
- Доступ к мессенджеру Телеграм.

11.2 Инструкция

На рисунке 18 представлен скриншот приветственного окна бота. Нажмите кнопку «Start», чтобы начать пользоваться ботом. После этого появится приветственное сообщение бота. Далее можно начать пользоваться ботом.

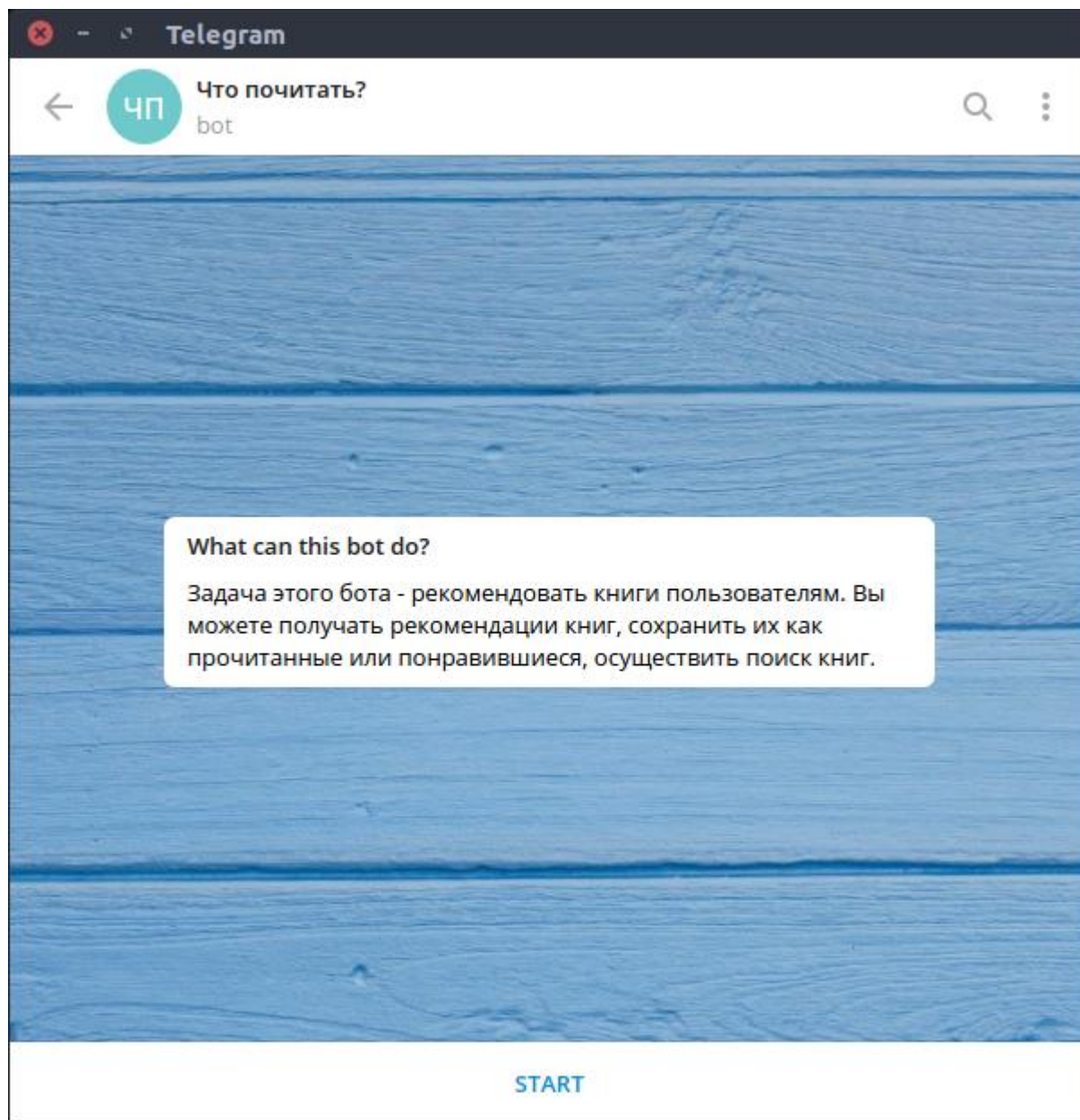


Рисунок 18

На рисунке 19 представлен скриншот сообщения бота после того, как пользователь нажмет кнопку «Start» на приветственном окне.

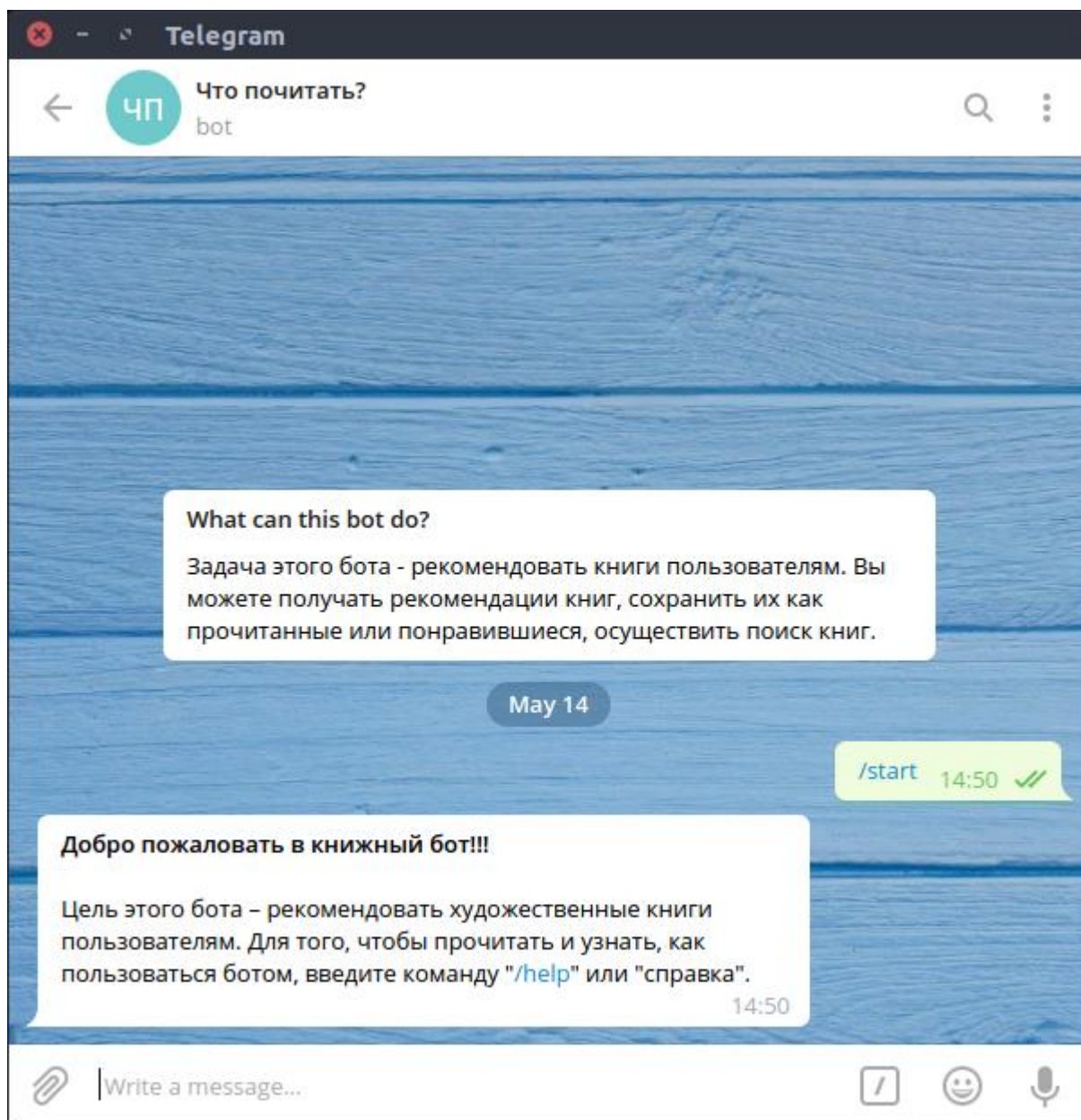


Рисунок 19

Чтобы посмотреть, какие команды можно отправить боту, введите символ «дробь» (слеш) «/» в форме для отправки сообщений и наверху формы будут показаны все доступные команды и краткое описание этих команд. На рисунке 20 представлен скриншот всех доступных команд бота.

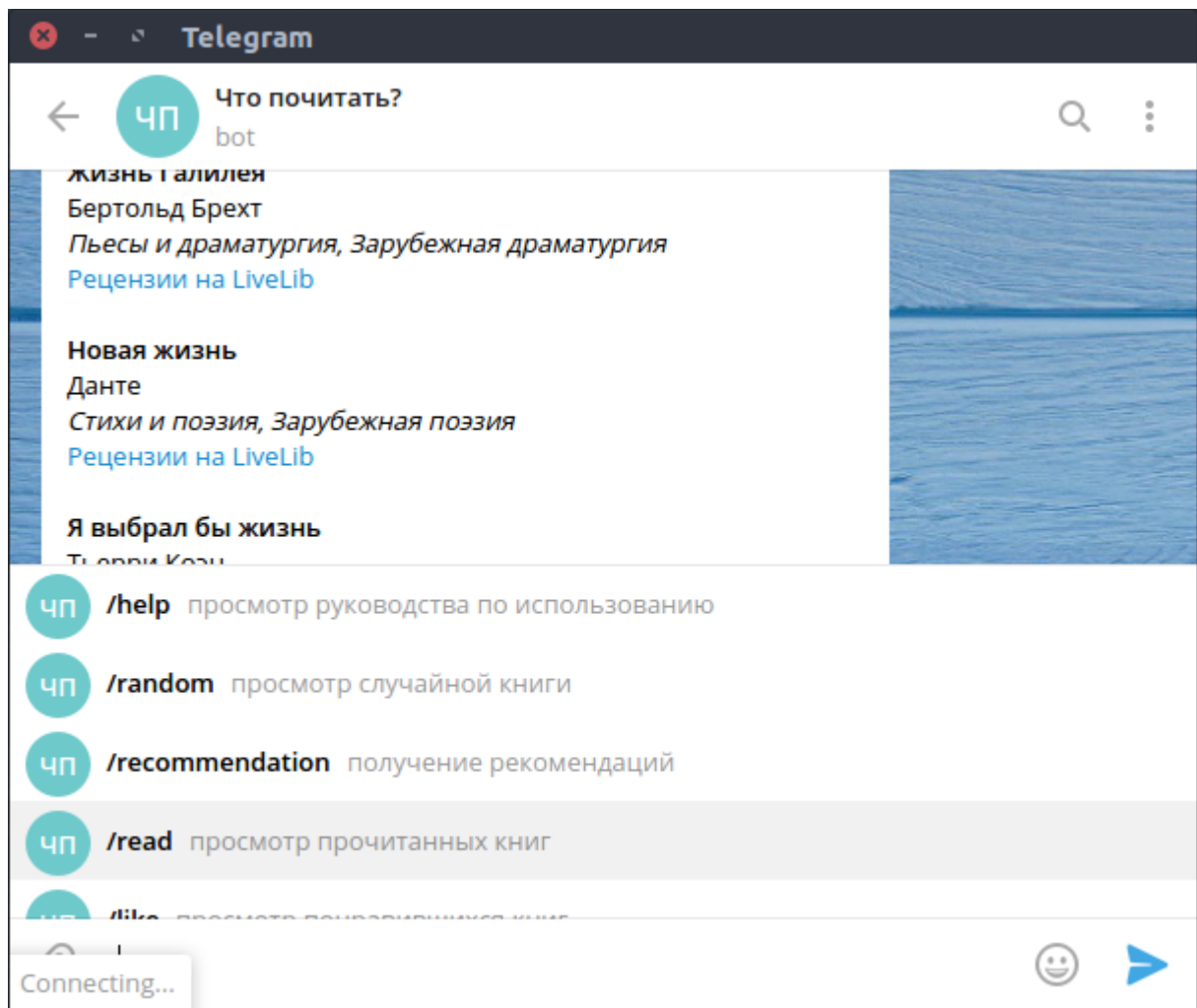


Рисунок 20

Пользователь может посмотреть руководство по использованию (справку), отправив боту сообщение «/help». Появится подробное описание всех доступных команд. На рисунке 21 представлен скриншот руководства по использованию бота.

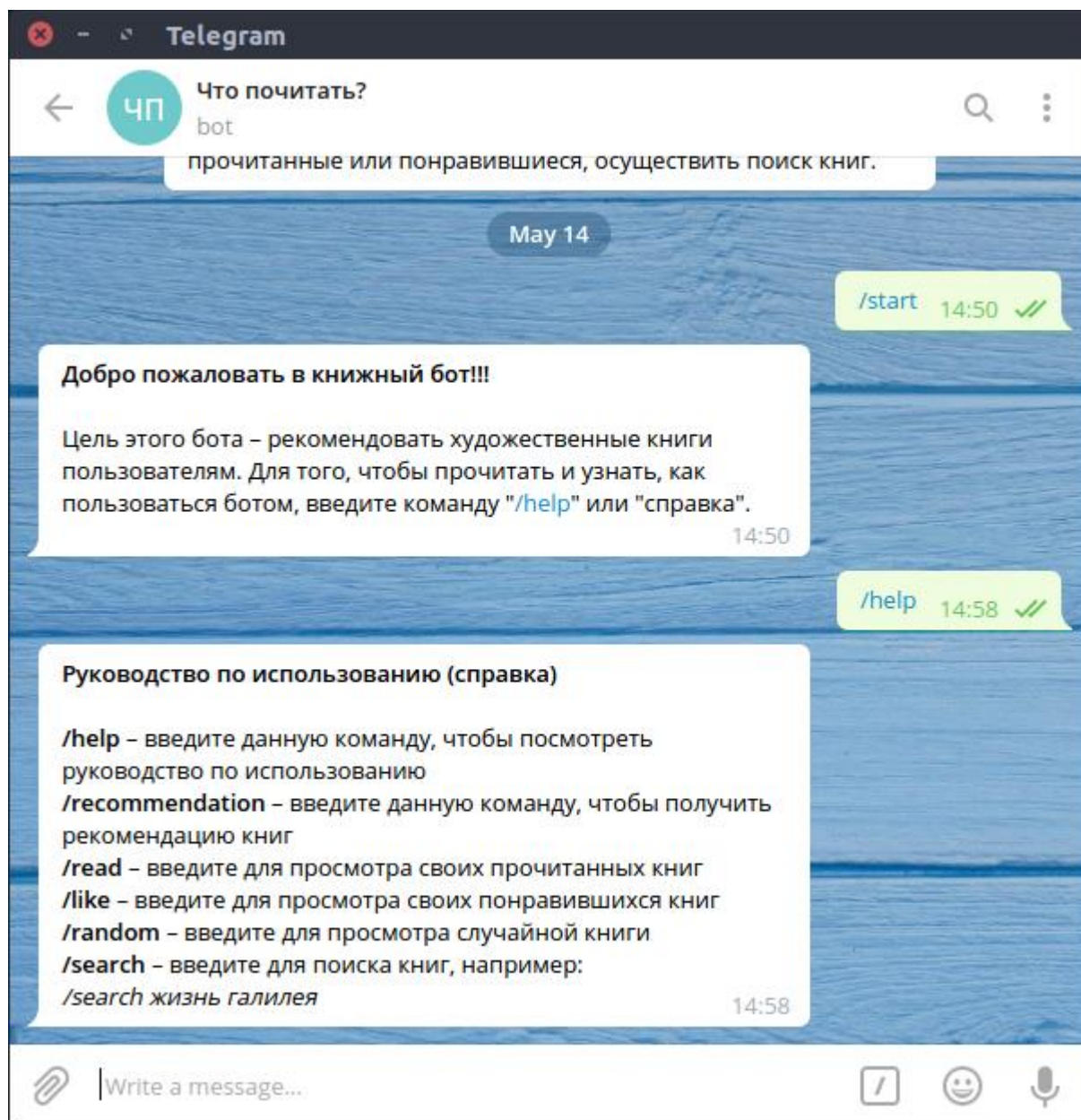


Рисунок 21

Чтобы получить рекомендацию, отправьте сообщение «/recommendation», далее на экране появится информация о рекомендуемой книге (название, автор, жанры, описание и ссылка на рецензии), а также три кнопки для выбора, чтобы рекомендательная система бота могла понять, какие книги предпочитает пользователь. На рисунке 22 представлен скриншот ответного сообщения бота на сообщение пользователя «/recommendation», иначе говоря, отображение рекомендуемой книги пользователю.

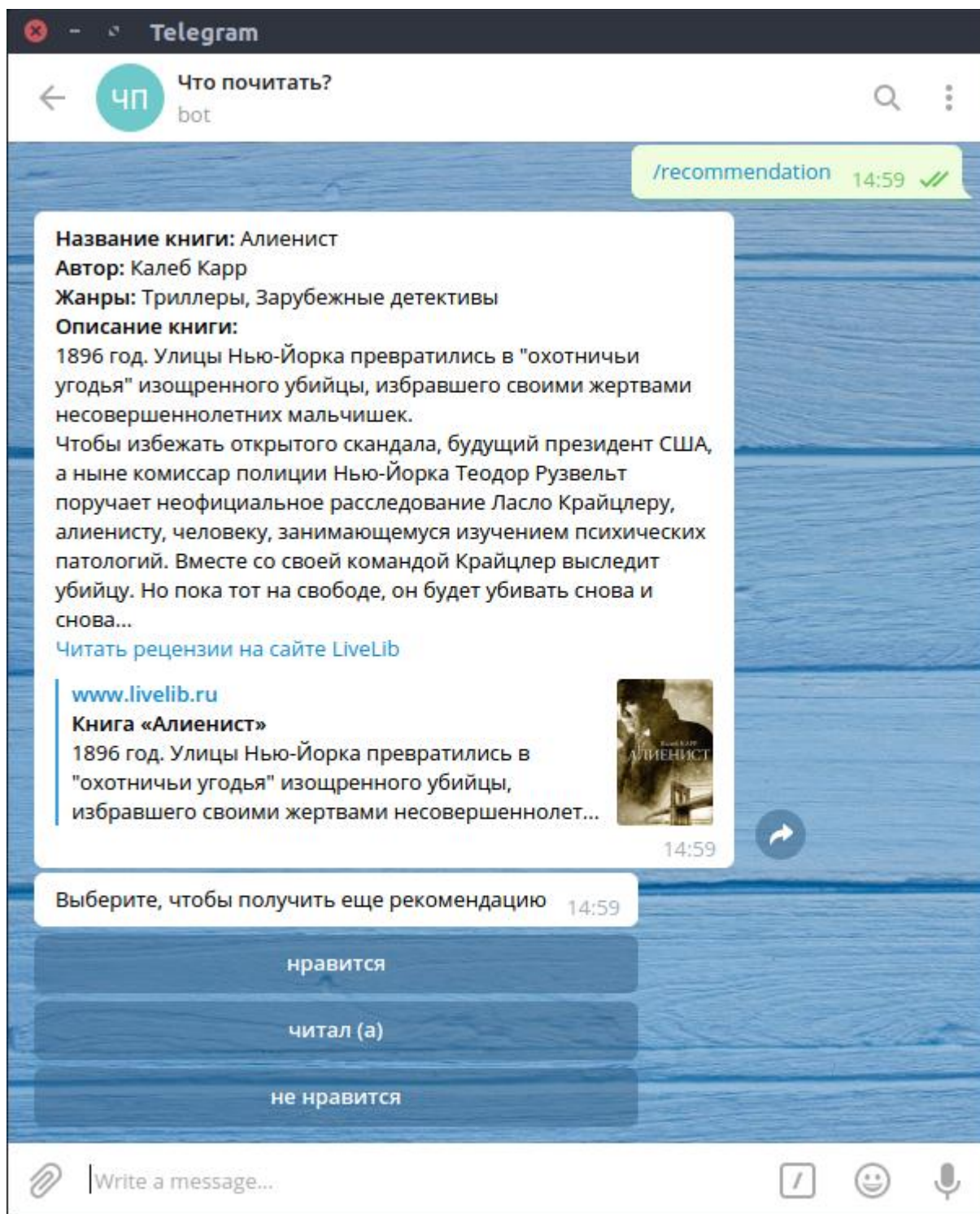


Рисунок 22

Пользователь может получить информацию о случайной книге, отправив сообщение «/random» боту. После того, как пользователь отправил сообщение «/random» боту, на экране будет показана информация о любой книге, находящейся на сервере. На рисунке 23 представлен скриншот ответного сообщения бота на сообщение пользователя «/random», иначе говоря, отображение случайной книги пользователю.

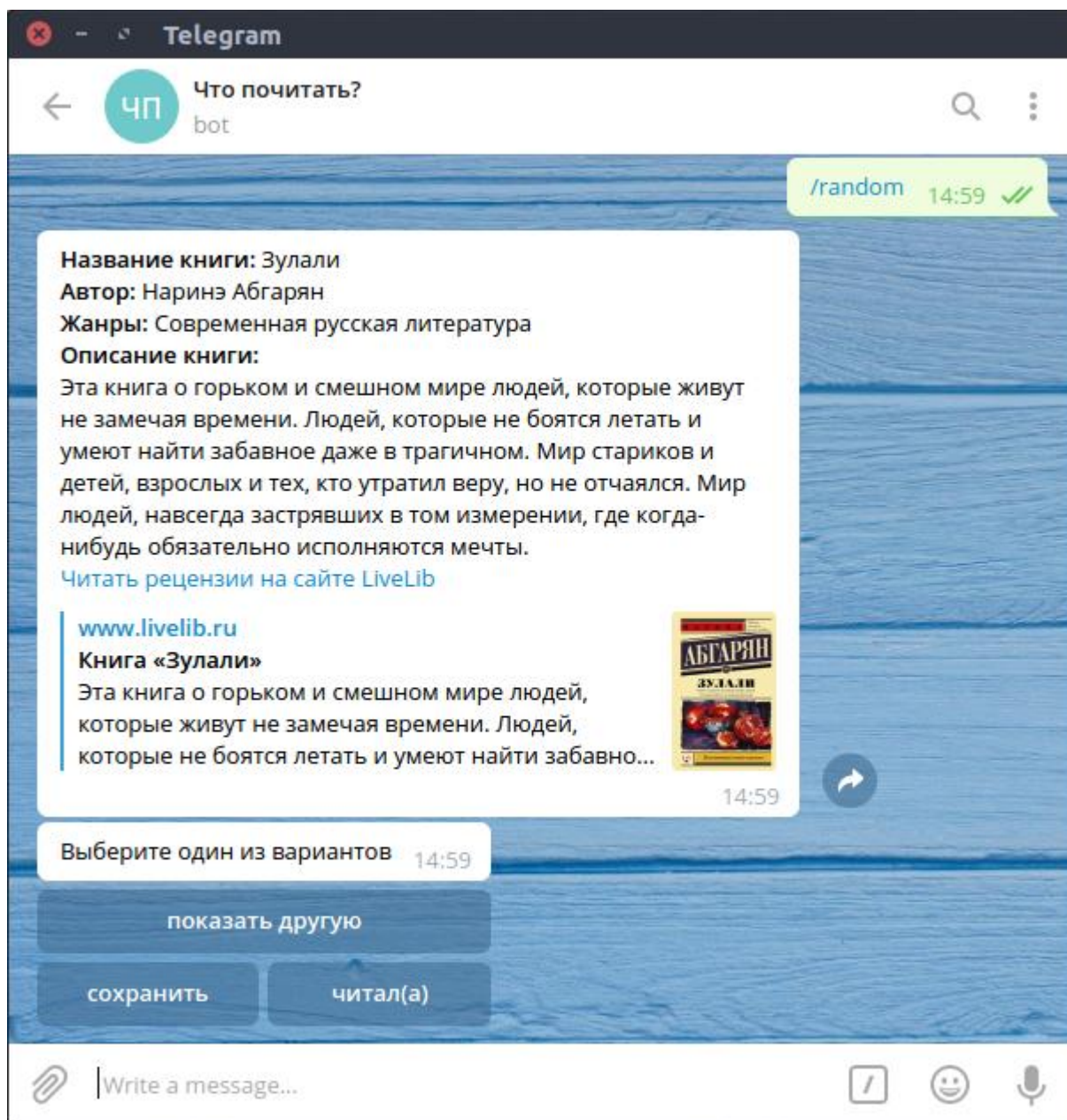


Рисунок 23

Чтобы посмотреть свои прочитанные или понравившиеся книги, введите «/read» или «/like» соответственно. На экране появится первая книга, которая была выбрана как прочитанная или понравившаяся соответственно. Чтобы посмотреть другую добавленную книгу нажмите кнопки «предыдущая» или «следующая». На рисунке 24 представлен скриншот отображения информации о прочитанной книге пользователя.

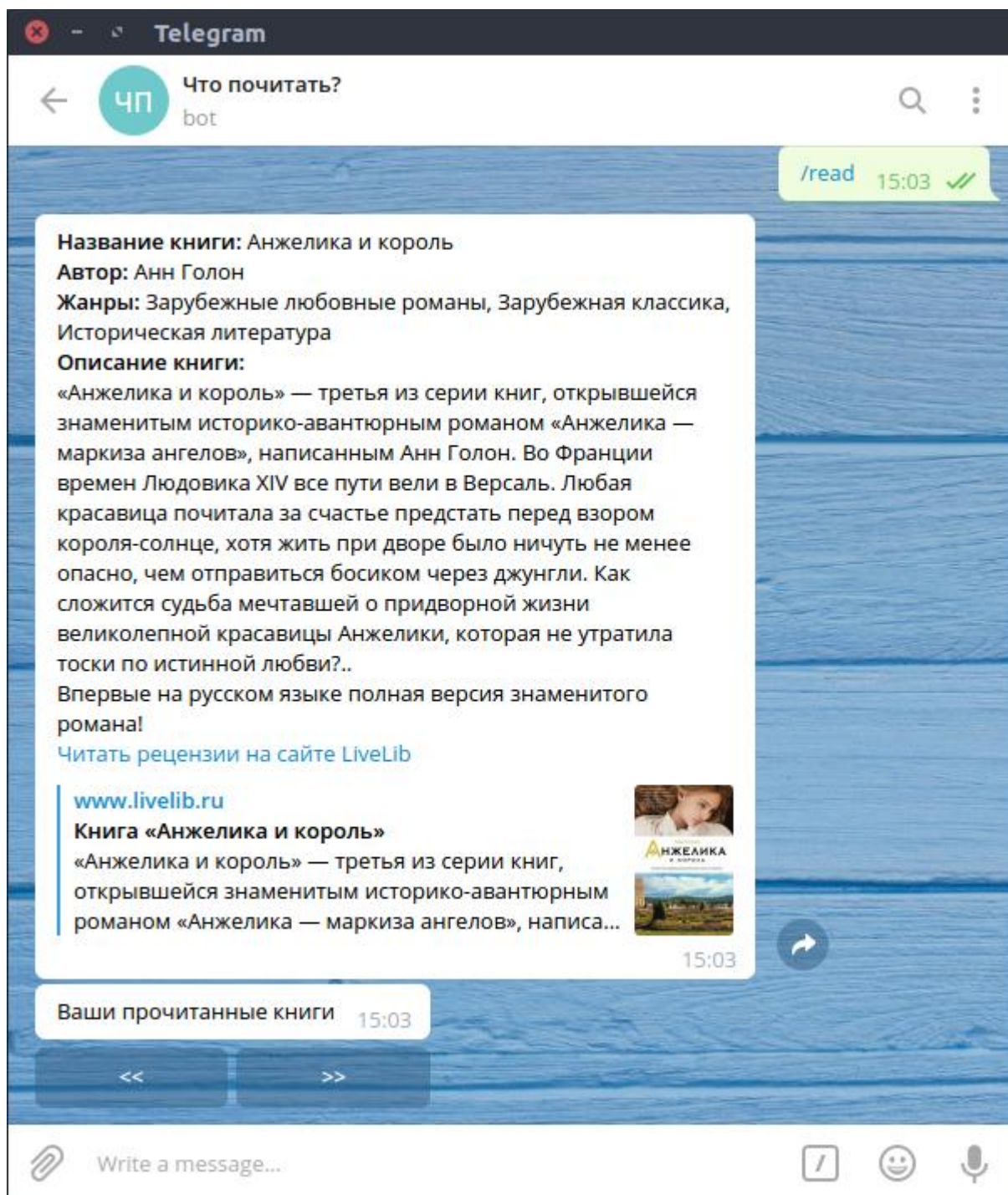


Рисунок 24

Чтобы осуществить поиск, отправьте боту сообщение «/search [название книги]», например, «/search Алиенист». На экране будут показана информация (название, автора, жанры, ссылка на рецензии) найденных книг. На рисунке 25 представлен скриншот отображения найденных книг при поиске.

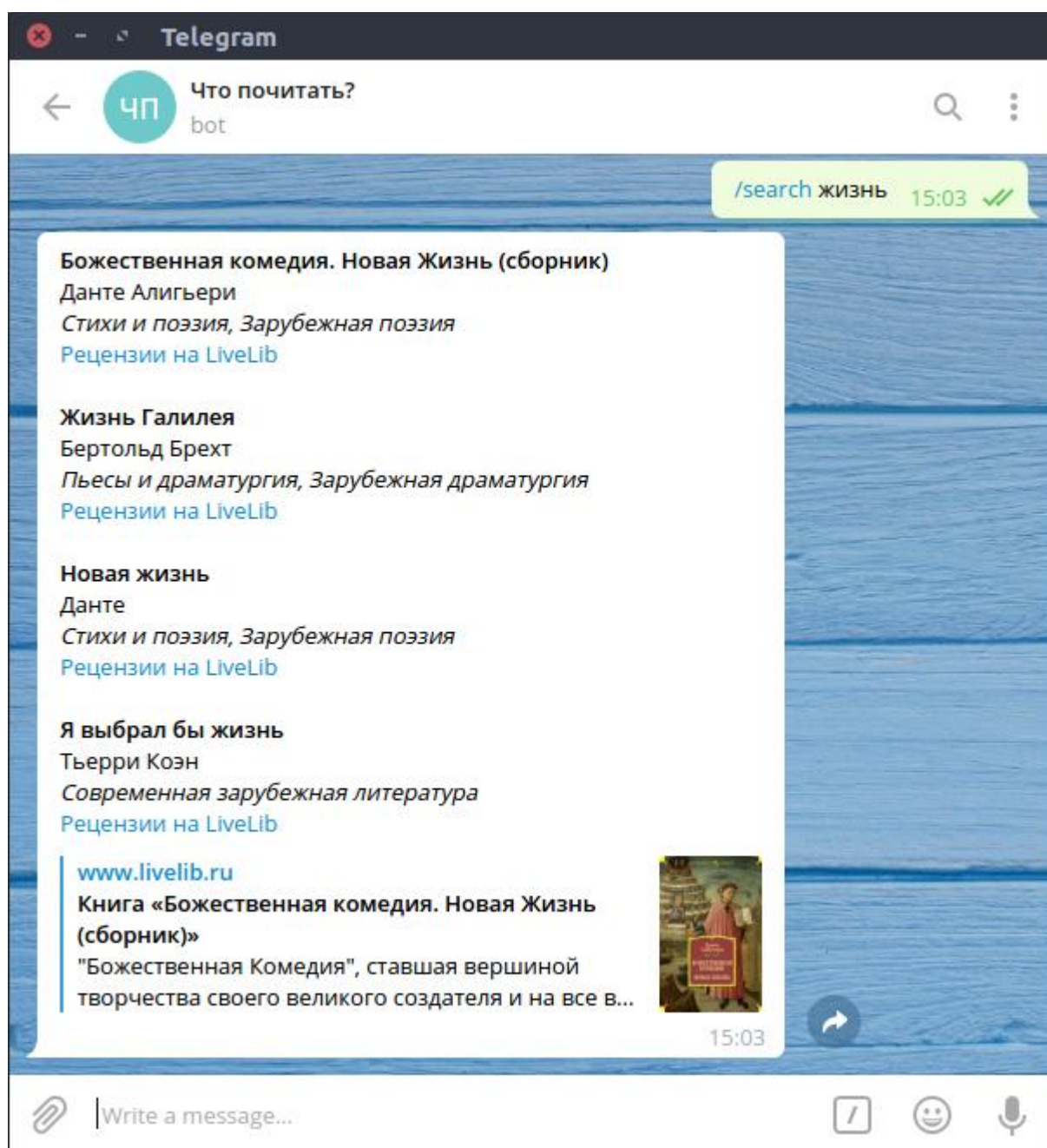


Рисунок 25

ЗАКЛЮЧЕНИЕ

Результатом выпускной квалификационной работы является разработанная система рекомендаций на основе собранной информации о книгах и отзывах о них, а также пользовательский интерфейс в виде чат-бота в мессенджере Телеграм.

В рамках выполнения выпускной квалификационной работы были изучены следующие технологии:

- Языки программирования PHP и JavaScript.
- Платформы Node.js и CakePHP.
- СУБД MySQL.
- Телеграм API.

В рамках выполнения выпускной квалификационной работы были получены следующие теоретические и практические знания:

- По сбору и обработке данных.
- По разработке алгоритмов рекомендательных систем.
- По разработке чат-ботов.

В рамках выпускной квалификационной работы были выполнены следующие виды работ:

- Проведено исследование предметной области.
- Сформулированы функциональные и нефункциональные требования.
- Выбраны инструменты и средства для проектирования и разработки.
- Спроектирована структура базы данных и выбрана архитектура для разработки рекомендательной системы и чат-бота.
- Разработаны модуль для сбора данных, рекомендательная система, а также пользовательский интерфейс в виде чат-бота.
- Выполнена оптимизация программного кода.
- Разработаны тестовые сценарии и тестовые процедуры для проверки корректности функционирования бота.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 EPICSTARS.com [Электронный ресурс] – URL:<https://ru.epicstars.com/boty-Telegram>, свободный. – Яз. рус., англ. – (Дата обращ. 05.02.2018).
- 2 LPGENERATOR.ru [Электронный ресурс] – URL:<https://lpgenerator.ru/blog/2015/12/25/rekomendatelnye-sistemy-chto-eto>, свободный. – Яз. рус., англ. – (Дата обращ. 05.02.2018).
- 3 LiveLib.ru [Электронный ресурс] – URL:<https://livelib.ru>, свободный. – Яз. рус., англ. – (Дата обращ. 09.01.2018).
- 4 LitRes.ru [Электронный ресурс] – URL:<https://litres.ru>, свободный. – Яз. рус., англ. – (Дата обращ. 09.01.2018).
- 5 GoodReads.com [Электронный ресурс] – URL:<https://goodreads.com>, свободный. – Яз. рус., англ. – (Дата обращ. 12.02.2018).
- 6 Имхонет [Электронный ресурс] – URL:<https://imhonet.ru>, свободный. – Яз. рус., англ. – (Дата обращ. 12.02.2018).
- 7 Habrahabr.ru [Электронный ресурс] – URL:<https://habrahabr.ru/post/214079>, свободный. – Яз. рус., англ. – (Дата обращ. 18.03.2018).
- 8 WikiGnome.org [Электронный ресурс] – URL:<https://wiki.gnome.org/Apps/Dia>, свободный. – Яз. рус., англ. – (Дата обращ. 18.03.2018).
- 9 MySQL [Электронный ресурс] – URL:<https://www.mysql.com/products/workbench>, свободный. – Яз. рус., англ. – (Дата обращ. 06.02.2018).
- 10 Node.js – Руководство для начинающих [Электронный ресурс] – URL:<https://nodeguide.ru/doc/felix/beginner>, свободный. – Яз. рус., англ. – (Дата обращ. 14.02.2018).

ПРИЛОЖЕНИЕ А

Листинг программного кода класса «CollectionBooksShell».

```
class CollectionBooksShell extends Shell {  
    /**  
     * main() method.  
     * @return bool|int|null Success or error code.  
     */  
    public function main() {  
        $this->collectionBooksByGenres();  
        //$this->addCollectionBook();  
    }  
    public function collectionBooksByGenres() {  
        $id = 14;  
        $livelib = LIVELIB;  
        $genre = $this->Genres->find('all')->where(['id' => $id])->first();  
        if (empty($genre)) {  
            $this->out('Запись не найдена');  
        } else {  
            $pageOfGenre = $genre['link_livelib'];  
            $dom = new \DOMDocument("1.0", "UTF-8");  
            $internalErrors = libxml_use_internal_errors(true);  
            $dom->loadHTMLFile($pageOfGenre);  
            $finder = new \DOMXPath($dom);  
            $bookName = "block-book-title";  
            $bookISBN = "book-details-info";  
            $query1 = sprintf("//a[contains(@class, '%s')]", $bookName);  
            $query2 = sprintf("//*[contains(@class, '%s')]", $bookISBN);
```

```

$books = $finder->query($query1);
$infos = $finder->query($query2);
for ($id = 0; $id < $books->length; $id++) {
    $isbn = $infos[$id]->nodeValue;
    $afterisbn = stristr($isbn, 'ISBN: ');
    if (!empty($afterisbn)) {
        $keywords = preg_split("/[\s,]+/", $afterisbn);
        $isbn = $keywords[1];
        $this->out($books[$id]->nodeValue . PHP_EOL);
        $this->out($livelib . $books[$id]->getAttribute('href') . PHP_EOL);
        $this->out($isbn . PHP_EOL);
        $checkisbn = $this->CollectionBooks->checkISBN($isbn);
        if (!empty($checkisbn)) {
            $this->out("Книга уже добавлена в таблицу");
        } else {
            $book = $this->CollectionBooks->newEntity();
            $book->name = $books[$id]->nodeValue;
            $book->isbn = $isbn;
            $book->link = $livelib . $books[$id]->getAttribute('href');
            $book->status = 'new';
            $status = $this->CollectionBooks->addBook($book);
            $this->out($status);
        }
    } else {
        continue;
    }
}
libxml_use_internal_errors($internalErrors);
}

```

```
}
```

```
public function addCollectionBook() {
```

```
    $id = 150;
```

```
    $addBook = array();
```

```
    $book = $this->CollectionBooks->find('all')->where(['id' => $id])->first();
```

```
    if (empty($book)) {
```

```
        $this->out('Запись не найдена');
```

```
    } else {
```

```
        if ($book['status'] == 'new') {
```

```
            $pageOfBook = $book['link'];
```

```
            $dom = new \DOMDocument("1.0", "UTF-8");
```

```
            $internalErrors = libxml_use_internal_errors(true);
```

```
            $dom->loadHTMLFile($pageOfBook);
```

```
            $finder = new \DOMXPath($dom);
```

```
            $reviewCount = "menu-tabs-reviews";
```

```
            $bookTitle = "name";
```

```
            $bookAuthor = "book-author";
```

```
            $bookCover = "main-image-book";
```

```
            $isbn = "isbn";
```

```
            $genres = "label-genre";
```

```
            $description = "description";
```

```
            $rating = "ratingValue";
```

```
            $name = $finder->query(sprintf("//*[contains(@itemprop, '%s')]",  
$bookTitle));
```

```
            $count = $finder->query(sprintf("//a[contains(@class, '%s')]",  
$reviewCount));
```

```

$author = $finder->query(sprintf("//a[contains(@id, '%s')]", $bookAuthor));
$cover = $finder->query(sprintf("//img[contains(@id, '%s')]", $bookCover));
$isbn = $finder->query(sprintf("//span[contains(@itemprop, '%s')]", $isbn));
$genres = $finder->query(sprintf("//a[contains(@class, '%s')]", $genres));
$description = $finder->query(sprintf("//p[contains(@itemprop, '%s')]",
$description));

$rating = $finder->query(sprintf("//span[contains(@itemprop, '%s')]",
$rating));

$reviewCount = $count->item(0)->nodeValue;
$count = stristr($reviewCount, ' ');
$count = $count;
$count = trim($count);
$cover = $cover->item(0)->getAttribute('src');

/* Собранные данные храним в массив, чтобы в дальнейшем сохранить в
базу */

$addBook['name'] = $name->item(0)->nodeValue;
$addBook['authors'] = $author->item(0)->nodeValue;
$addBook['cover'] = $cover;
$addBook['genres'] = "";
$addBook['reviews_count'] = $count;
$addBook['isbn'] = $isbn->item(0)->nodeValue;
$addBook['description'] = $description->item(0)->nodeValue;
$addBook['rating'] = $rating->item(0)->nodeValue;
if (empty($addBook['reviews_count']))
    $addBook['reviews_count'] = 0;
$addBook['link'] = $book['link'];
foreach ($genres as $genre) {
    $addBook['genres'] .= $genre->nodeValue . ", ";
}

```

```

$addBook['genres'] = trim($addBook['genres'], " ,");
$addBook['description'] = trim($addBook['description']);
$addBook['fromsite'] = 'livelib';
/* Конец создания массива */

if ($addBook['reviews_count'] > 1) {
    $checkisbn = $this->Books->getBookByISBN($addBook['isbn']);
    if (!empty($checkisbn)) {
        $this->out("Книга уже добавлена в БД");
    } else {
        $added_status = $this->Books->addBook($addBook);
        if ($added_status) {
            $added = $this->CollectionBooks->changeBookStatus($book);
            $this->out("Книга добавлена");
        } else {
            $this->out("Ошибка добавления");
        }
    }
} else {
    $this->out("Книге оставлены очень мало рецензий");
}

libxml_use_internal_errors($internalErrors);
} else {
    $this->out("Книга уже добавлена в БД");
}
}
}
}

```