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

ОТЧЕТ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

ЭТАП: ПМ08 (ПМ06) Участие в анализе и разработке требований к программному продукту

ТЕМА ПРАКТИКИ: Разработка требований к программному продукту «Книжный бот рекомендации книг»

Директор компании Потапенко В. А.

E2E4GU.RU «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2018

Студент 4 курса Дугуржап Х. Ш.

гр. 403с «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2018

Новосибирск

2018

СОДЕРЖАНИЕ

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ](#_gjdgxs) 2

[ВВЕДЕНИЕ](#_30j0zll) 3

[1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ](#_1fob9te) 5

[2 ПОСТАНОВКА ЗАДАЧИ](#_3znysh7) 6

[3 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ](#_2et92p0) 8

[4 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ](#_tyjcwt) 13

[Требования к программному обеспечению](#_3dy6vkm) 14

[Требования к аппаратному обеспечению](#_1t3h5sf) 14

[Требования к надежности](#_4d34og8) 14

[Требования к безопасности](#_2s8eyo1) 15

[ЗАКЛЮЧЕНИЕ](#_17dp8vu) 15

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ](#_3rdcrjn) 16

# ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

* API (Application Programming Interface, программный интерфейс приложения) – это набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах. Используется программистами при написании всевозможных приложений.
* SSL (Secure Sockets Layer, уровень защищённых сокетов) – это цифровая подпись сайта, которая нужна для работы протокола защищенной передачи данных в сети. Проще говоря, она обеспечивает шифрованное соединение между пользователем и сайтом. То есть информация, которой они обмениваются, защищена от посторонних — провайдера, оператора, администратора вайфай-сети и прочих.
* ISBN (International Standard Book Number) – Международный стандартный книжный номер.
* БД – база данных.
* XSS (Cross-Site Scripting, межсайтовый скриптинг) – 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.

# ВВЕДЕНИЕ

Темой данной выпускной квалификационной работы является разработка чат-бота «Книжный бот рекомендации книг» для мессенджера Telegram. Рекомендации позволят лучше ориентироваться в популярной литературе, сократить время на поиск книги. На основе этой задачи были сформулированы следующие функциональные требования:

* Бот рекомендует книги пользователям на основе собранных данных о рецензиях различных книг и отправляет уведомления;
* Бот также сохраняет прочитанные и понравившиеся книги;
* Бот также позволяет осуществить поиск книг;

Данный бот рассчитан на аудиторию пользователей, увлекающихся художественной литературой.

Виды проделанных работ:

* исследование предметной области;
* сбор и анализ функциональных и нефункциональных требований;
* изучение и разработка механизма рекомендательной системы;
* выбор инструментов и средств разработки;
* проектирование архитектуры БД;
* проектирование архитектуры чат-бота;
* тестирование и отладка чат-бота.

Используемые программные средства и технологии:

* язык программирования PHP не ранее версии 7.0;
* база данных MySQL;
* сервер с SSL-сертификатом Heroku для хранения чат-бота;
* Node.JS TelegramBot API.

# 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Telegram — 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 режиме диалога предоставляют и собирают данные у пользователей, инициируют различные процессы во внешних системах.

В Telegram используется один общий вид ботов, которых от обычных пользователей отличает только наличие приставки «bot» в имени. Сами же боты делятся на несколько направлений:

* Чат-боты — представляют из себя простейший чат, 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-асcистенты — боты, разработанные различными онлайн-сервисами как дополнение к основной веб-версии.

Четкого деления нет, так как некоторые боты содержат сразу несколько механик и успешно выполняют множество пользовательских задач. С их помощью можно переводить, обучаться, тестировать, искать информацию, играть в игры и даже пользоваться другими сервисами и взаимодействовать с вещами, имеющими выход в глобальную сеть[1].

# 2 ПОСТАНОВКА ЗАДАЧИ

Цель работы — проектирование и разработка Telegram-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 авторам, названиям и жанрам;
* составление списка прочитанных книг (на их основе также дают рекомендации);
* поиск сайтов для покупки книг.

Рекомендательная система собирает рецензии людей на различные книги с сайтов LiveLib (www.livelib.ru) и ЛитРес (www.litres.ru), 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: разработать систему и реализовать её интерфейс с помощью telegram-бота. Система предназначена для того, чтобы рекомендовать книги пользователям, а также для осуществления поиска книг по авторам, названиям, жанрам и для сохранения любимых и прочитанных книг в закладки. Название, автор, ISBN, обложка и прочая информация сохраненных книг хранится в БД. Задача была поставлена научным руководителем.

# 3 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

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

* просмотр рекомендаций;
* получать уведомления о рекомендуемых книгах;
* Просмотр книг, добавленных в прочитанные или понравившиеся;
* поиск книг по авторам, названиям, жанрам;
* просмотр справки.

Кроме бота, нужно создать рекомендательную систему книг. Рекомендательная система собирает рецензии различных книг. Сбор рецензий осуществляется по сайтам: LiveLib ([www.livelib.ru](http://www.livelib.ru)) и ЛитРес ([www.litres.ru](http://www.litres.ru)). После сбора рецензий, система группирует книги по следующим критериям: оставили рецензию книгам или поставили максимальную оценку один и тот же человек. Затем система будет отправлять рекомендации книг пользователям бота. Каким образом будет собирать рецензии и отправлять рекомендации, подробно показаны в вариантах использования №1 и №2.

Ниже показаны все варианты использования ботом и его реакцию на совершенные действия пользователя. В вариантах использования имеются следующие роли, которые выполняют те или иные действия:

* Пользователь — человек, который пользуется ботом;
* Бот-telegram — бот, реализующий интерфейс рекомендательной системы;
* Система — рекомендательная система.

# Варианты использования

Use Case #1.1

Цель: сбор рецензий

1. Система находит книгу M с сайтов LiveLib ([www.livelib.ru](http://www.livelib.ru)) или ЛитРес ([www.litres.ru](http://www.litres.ru)) и все рецензии на эту книгу;

2. Система ищет рецензии пользователей, которые поставили высокий балл этой книге;

3. Система зафиксирует рецензию N-го пользователя, который поставил высокий балл книге M;

4. Далее начинается Use Case #1.2

Use Case #1.2

Цель: сбор рецензий

1. Система собирает рецензию N-го рецензента на книгу M с сайтов LiveLib ([www.livelib.ru](http://www.livelib.ru)) или ЛитРес (www.litres.ru);

2. Система собирает рецензию N+1-го рецензента на книгу M с сайтов LiveLib ([www.livelib.ru](http://www.livelib.ru)) или ЛитРес (www.litres.ru);

3. Система анализирует, какую оценку поставили данные рецензенты на книгу M;

4. Если оба рецензента поставили высокую оценку на эту книгу, то система объединяет этих рецензентов в одну группу;

5. Система находит рецензии на разные книги с высокими оценками N-го и N+1-го пользователя;

6. Система объединяет все книги в группу G, на которой писали рецензии и поставили высокий балл N-ый и N+1-ый рецензент;

7. Система объединяет эти книги и сохраняет название, автор, год издания, ISBN, краткое описание этих книг. Если пользователь бота выберет, что книга M нравится, то система будет рекомендовать книги из группы G.

Use Case #1.3

Цель: сбор рецензий

1. Система собирает рецензию N-го рецензента на книгу M с сайтов LiveLib ([www.livelib.ru](http://www.livelib.ru)) или ЛитРес (www.litres.ru);

2. Система собирает рецензию N+1-го рецензента на книгу M с сайтов LiveLib ([www.livelib.ru](http://www.livelib.ru)) или ЛитРес (www.litres.ru);

3. Система анализирует, какую оценку поставили данные рецензенты на книгу M;

4. Если рецензенты поставили разные оценки на эту книгу, то система ищет другого N+1-го рецензента, который поставил высокую оценку на эту книгу;

5. Система находит другого рецензента и также анализирует оценку этого рецензента;

Use Case #2.1

Цель: получений рекомендаций

1. Пользователь вводит команду "рекомендация";

2. бот показывает рекомендуемую книгу системы и кнопки «нравится», «прочитано» и «дальше»;

3. пользователь выбирает вариант «нравится»;

4. бот добавляет книгу в понравившиеся книги и показывает следующую книгу, похожую на первую книгу и те же кнопки.

Use Case #2.2

Цель: получений рекомендаций

1. Пользователь вводит команду «рекомендация»;

2. бот показывает рекомендуемую книгу и кнопки «нравится», «прочитано» и «дальше»

3. пользователь выбирает вариант «прочитано»;

4. бот добавляет книгу в прочитанные книги и показывает следующую книгу, похожую на первую книгу и те же кнопки.

Use Case #2.3

Цель: получений рекомендаций

1. Пользователь вводит команду «рекомендация»

2. бот показывает рекомендуемую книгу и кнопки «нравится», «прочитано» и «дальше»

3. пользователь выбирает вариант «дальше»

4. бот показывает следующую книгу, не похожую на первую книгу и те же кнопки

Use Case #3.1

Цель: получение уведомлений о рекомендуемых книгах

1. Пользователь вводит команду «уведомления»

2. бот показывает список пунктов, сколько раз в день пользователь хочет получить рекомендации:

\* 1 книга раз в день

\* 2 книги раз в день

\* 1 книга раз в неделю

\* 5 книг раз в неделю (каждое воскресенье)

\* 10 книг раз в месяц (в конце каждого месяца)

\* не получать уведомления;

3. пользователь выбирает один из пунктов, бот будет отправлять уведомления в соответствии с выбором пользователя.

Use Case #3.2

Цель: пользователь только запустил бота и еще не подписался на уведомления

1. Бот не будет отправлять никаких уведомлений о рекомендуемых книгах системы, пользователь может просмотреть рекомендуемые книги, набрав команду «рекомендация» (UseCase #1), а чтобы подписаться на уведомления, вводить команду «уведомления» (UseCase #2.1)

Use Case #3.3

Цель: пользователь подписался на уведомления и хочет отписаться от уведомлений

1. Пользователь вводит команду «уведомления»;

2. бот показывает список пунктов, сколько раз в день пользователь хочет получить рекомендации, среди них есть пункт "не получать уведомления" (UseCase #2.1, этап 2);

3. пользователь выбирает пункт «не получать уведомления» и бот перестает отправлять уведомления, чтобы заново подписаться на уведомления, пользователю нужно вводить команду «уведомления».

Use Case #4.1

Цель: просмотр книг, добавленных в «прочитанные»

1. Пользователь вводит команду «прочитанные»;

2. бот выводит список книг, добавленных в прочитанные;

3. пользователь выбирает одну из книг;

4. бот показывает информацию о книге: название, автор, год издания, ISBN, краткое описание, жанр.

Use Case #4.2

Цель: просмотр книг, добавленных в «понравившиеся»

1. Пользователь вводит команду «понравившиеся»;

2. бот выдает список книг, добавленных в понравившихся;

3. пользователь выбирает одну из книг;

4. бот показывает информацию о книге: название, автор, год издания, ISBN, краткое описание, жанр.

Use Case #4.3

Цель: просмотр книг, добавленных в «понравившиеся» или «прочитанные» (не найдены книги)

1. Пользователь вводит команды «понравившиеся» или «прочитанные»;

2. бот выводит сообщение, что пока нет прочитанных или понравившихся книг.

Use Case #5.1

Цель: поиск книг по названию

1. Пользователь вводит команду «поиск»;

2. бот выдает справку о том, какие команды нужно ввести, чтобы искать книги:

\* поиск по авторам: «автор {{имя автора}}»

\* поиск по жанрам «жанр {{название жанра}}»

\* поиск по названиям «книга {{название книги}}»;

3. пользователь вводит «книга {{название книги}}»;

4. бот выдает список всех книг, где в названии {{название книги}};

5. пользователь выбирает нужную книгу и заходит в страницу книги;

6. на странице книги показана следующая информация – название книги, автор, год издания, ISBN, краткое описание, а также кнопки «прочитано», «нравится», «купить»

Use Case #5.2

Цель: поиск книг по авторам

1. Пользователь вводит команду «автор {{имя автора}}»;

2. бот выдает список всех книг автора {{имя автора}}

3. пользователь выбирает нужную книгу и заходит в страницу книги;

4. на странице книги показана следующая информация: название книги, автор, год издания, ISBN, краткое описание, кнопки «прочитано», «нравится», «купить».

Use Case #5.3

Цель: поиск книг по жанрам

1. Пользователь вводит команду «жанр {{название жанра}}»;

2. бот выдает список всех книг жанра {{жанр}};

3. пользователь выбирает нужную книгу и заходит в страницу книги;

4. на странице книги показана следующая информация - название книги, автор, год издания, ISBN, краткое описание, кнопки "прочитано", «нравится», «купить».

Use Case #5.4

Цель: поиск книг (ошибка при поиске)

1. Пользователь вводит команду «книга {{...}}» или «автор {{...}}» или «жанр {{...}}»;

2. бот выдает, что результат не найден.

Use Case #6.1

Цель: просмотр справки

1. Пользователь вводит команду «справка»;

2. бот показывает всю справку, все команды бота и краткое описание.

# 4 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

# Требования к программному обеспечению

Для обеспечения функционирования программного средства необходимо использовать следующие программные средства:

* базу данных MySQL версии не ниже 5.7;
* интерпретатор PHP версии не ниже 7.0;
* Framework CakePHP версии не ниже 3.5;
* Node.JS Telegram Bot API;
* Amazon API;
* GoodReads API.

# 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 девайсу клиента:

* установленный мессенджер Telegram.

# 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.

# Требования к безопасности

Для обеспечения безопасности бота необходимы следующие требования:

* бот должен храниться в сервере с SSL-сертификатом для обеспечения шифрования;
* система должна обеспечивать защиту от XSS и SQL-инъекций.

# ЗАКЛЮЧЕНИЕ

В рамках производственной практики "Участие в анализе и разработке требований к программному продукту" были выполнены следующие пункты:

* Проведено исследование предметной области.
* Сформулированы функциональные и нефункциональные требования к боту.
* Выбраны инструменты и средства проектирования, разработки бота.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Боты в Telegram и как они работают - https://ru.epicstars.com/botyi-telegram