

Министерство образования и науки Российской Федерации

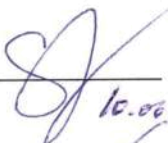
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СОВРЕМЕННОГО
КОМПЛЕКСНОГО ПРОГРАММНОГО РЕШЕНИЯ НА ПРИМЕРЕ
СЕРВИСА ПОМОЩИ АВТОЛЮБИТЕЛЯМ
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНИИТ
Шведова Вадима Валерьевича

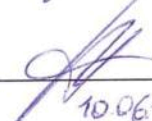
Научный руководитель
доцент, к.ф.-м.н.



10.06.17

В. Г. Самойлов

Заведующий кафедрой
к.ф.-м.н., доцент



10.06.17

С. В. Миронов

Саратов 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Основное содержание работы	4
2 Теоретическая часть	6
2.1 Веб-сервисы	6
2.1.1 REST технология	6
2.2 База данных	6
2.2.1 PostgreSQL	7
2.2.2 Логическая модель	7
2.3 Выбор среды разработки	7
2.3.1 IntelliJ IDEA	9
2.3.2 Android Studio	9
2.4 Необходимые технологии	9
2.4.1 Maven	10
2.4.2 Spring Framework	10
2.4.3 Spring MVC	10
2.4.4 Spring Security	10
2.4.5 Spring JdbcTemplate	11
2.4.6 Spring Boot	11
2.4.7 Angular JS	11
2.4.8 Bootstrap	11
2.4.9 Android	12
2.4.10 Apache HttpClient	12
2.4.11 Система автоматической сборки Gradle	12
2.4.12 Google Maps API	12
ЗАКЛЮЧЕНИЕ	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	15

ВВЕДЕНИЕ

В современном мире для создания комплексного программного решения часто прибегают к сервис-ориентированной архитектуре или к разработке веб-сервиса.

Популярность веб-сервисов, предоставляющих возможность пользователям быстро и удобно использовать множество услуг с помощью специальных мобильных приложений или окон браузера, все сильнее растет с каждым днем. Они предоставляют возможность одним программам обмениваться данными с другими.

К моменту появления понятия «веб-сервисы», в мире уже существовали технологии, позволяющие приложениям взаимодействовать на расстоянии, где одна программа могла вызвать определенный метод в другой программе, которая при этом могла быть запущена на компьютере, расположенном в другом городе или даже стране. Но это взаимодействие часто ограничивалось платформами и специфичными сетевыми протоколами.

По сути, веб-сервисы — это реализация абсолютно четких интерфейсов обмена данными между различными приложениями, которые написаны не только на различных языках, но и распределены на разных узлах сети.

Целью данной работы является изучить концепции веб-сервисной технологии и разработать современное комплексное программное обеспечение, которое удовлетворяет определенным требованиям к бизнес логике, к качеству эксплуатации и к удобству в использовании конечным пользователем.

Для реализации проекта необходимо выполнить следующие задачи:

- составить техническое задание;
- построить логическую и физическую модель базы данных;
- выбрать необходимые инструменты для разработки веб-сервисов и составления баз данных;
- спроектировать и разработать простую и масштабируемую архитектуру для серверной, клиентской и административной части.

1 Основное содержание работы

Основное направление разрабатываемого сервиса — это агрегирование различных сервисов помощи автолюбителям в одно мобильное приложение, предоставляя пользователю различные скидки на их услуги, возможность поиска ближайших сервисов через интерактивную карту, поиск текущих акций по категориям через меню мобильного приложения и возможность отправления заявок на предоставление определенных услуг.

В первом разделе составлено техническое задание и все необходимые требования к функционалу разрабатываемого сервиса.

Второй раздел носит теоретический характер, он посвящен теме веб-сервисов и в нем рассматриваются требуемые для решения задачи технологии и инструменты. Среди которых:

- СУБД Postgres;
- две среды разработки:
 - IntelliJ IDEA;
 - Android Studio;
- автоматический сборщик проекта Maven;
- Spring Framework и его компоненты:
 - Spring MVC;
 - Spring Security;
 - Spring JdbcTemplate;
 - Spring Boot;
- Angular JS;
- Bootstrap;
- Android;
- Apache HttpClient;
- Система автоматической сборки Gradle;
- Google Maps API.

Помимо этого в теоретическом разделе составляется логическая модель базы данных, которая удовлетворяет составленному техническому заданию.

В третьем разделе продемонстрированы все этапы разработки требуемого программного обеспечения.

Работа проходила в несколько этапов разработки:

- серверная часть — представляет собой java enterprise edition с примене-

нием фреймворка Spring, он отвечает обработке всех HTTP-запросов согласно REST-архитектуре от клиентов и панели администратора, обеспечивает авторизацию и общение с базой данных;

- клиенты — представляют собой нативные Android приложения, которые общаются с сервером путем REST-запросов;
 - панель администратора — веб-приложение для бизнес-пользователей, написанное на языке JavaScript с применением фреймворка AngularJS, которое также взаимодействует с сервером путем отправки REST-запросов.
- Для того чтобы реализовать серверную часть нужно решить следующие

основные задачи:

- составить физическую модель базы данных, которая соответствует требованиям логической модели;
- заполнить базу тестовыми данными;
- спроектировать REST API сервиса;
- сопоставить сущности базы данных с POJO сущностями;
- организовать доступ к базе данных из Java кода;
- реализовать защиту персональных данных для пользователей и для бизнес пользователей.

В заключении представлены основные выводы над проделанной работой. Список использованных источников содержит 26 наименований. В приложении представлено часть кода, который описывается в практической части, а также описание содержания компакт-диска, который и содержит исходный код всего проекта.

2 Теоретическая часть

2.1 Веб-сервисы

Термин «веб-сервис» имеет множество различных и расплывчатых определений, но на практике оно должно быть достаточным для понимания и написания кода, который состоит из сервера и клиента, также известного как consumer или requester, что хорошо описывается книгах М. Калина «Java Web Services: Up and Running» [1] и Т.С. Машнина «Web-сервисы Java» [2]. Как следует из названия, веб-сервис представляет собой своего рода сетевое приложение, которое обычно поставляется через HTTP протокол (HyperText Transport Protocol) или HTTPS (HTTP Secure).

Веб-сервисы могут быть написаны почти на любом языке программирования. Клиент веб-сервиса должен выполняться на устройстве, который имеет сетевой доступ, обычно через HTTP, к веб-серверу, на котором запущен сам веб-сервис. В более технических терминах веб-служба представляет собой распределенную программную систему, компоненты которой могут быть развернуты и выполняться на физически различных устройствах.

2.1.1 REST технология

REST — это архитектурный стиль для проектирования распределенных систем. Он не является стандартом, но определяет ограничения, такие как отсутствие состояний, клиент-серверная взаимосвязь и унифицированный интерфейс. REST не связан строго с HTTP, но чаще всего его ассоциируют именно с ним.

1. Ресурсы позволяют легко понять структуру каталогов URI.
2. Представления передают JSON или XML в качестве представления данных объекта и атрибутов.
3. Сообщения используют HTTP методы явно (например GET, POST, PUT и DELETE).
4. Отсутствие состояния взаимодействий не сохраняет контекст клиента на сервере между запросами.

2.2 База данных

Согласно информации из книги Н. П. Стружкина «Базы данных: Проектирование» [3], базы данных являются одним из основных компонентов почти

любой современной информационной системы. Это информационные структуры, содержащие взаимосвязанные данные о реальных объектах.

Особенностями такой совокупности данных являются:

- достаточно большие объемы информации;
- максимально возможная компактность хранения данных;
- возможность извлечения из базы данных разнообразной информации в определенной предметной области;
- удобные для пользователя вид и форма извлекаемой информации;
- высокая скорость доступа к данным;
- надежность хранения информации и возможность предоставления санкционированного доступа к данным для отдельных пользователей;
- удобство и простота конструирования пользователем запросов, форм и отчетов для выборки данных.

2.2.1 PostgreSQL

При планировании архитектуры сервиса помощи автолюбителям, для хранения и управления данными выбор пал на PostgreSQL.

PostgreSQL — это объектно-ориентированная система управления реляционными базами данных. Это означает, что это система для управления данными, которые хранятся в виде отношений. Отношение — это математический термин для таблицы. Понятие хранения данных в таблицах является сегодня достаточно тривиальным, что оно может показаться самоочевидным, однако есть несколько других способов организации баз данных. Файлы и каталоги в Unix-подобных операционных системах являются примером иерархической базы данных. Одно из наиболее современных направлений разработки СУБД — это объектно-ориентированные базы данных.

2.2.2 Логическая модель

В соответствии с техническим заданием логическая модель, удовлетворяющая всем требованиям сервиса представлена на рис. 1.

2.3 Выбор среды разработки

Выбор среды разработки — один из важных этапов при проектировании программного обеспечения. В рамках сервиса помощи автолюбителям для серверной части и панели администратора была выбрана IDE «IntelliJ IDEA»,

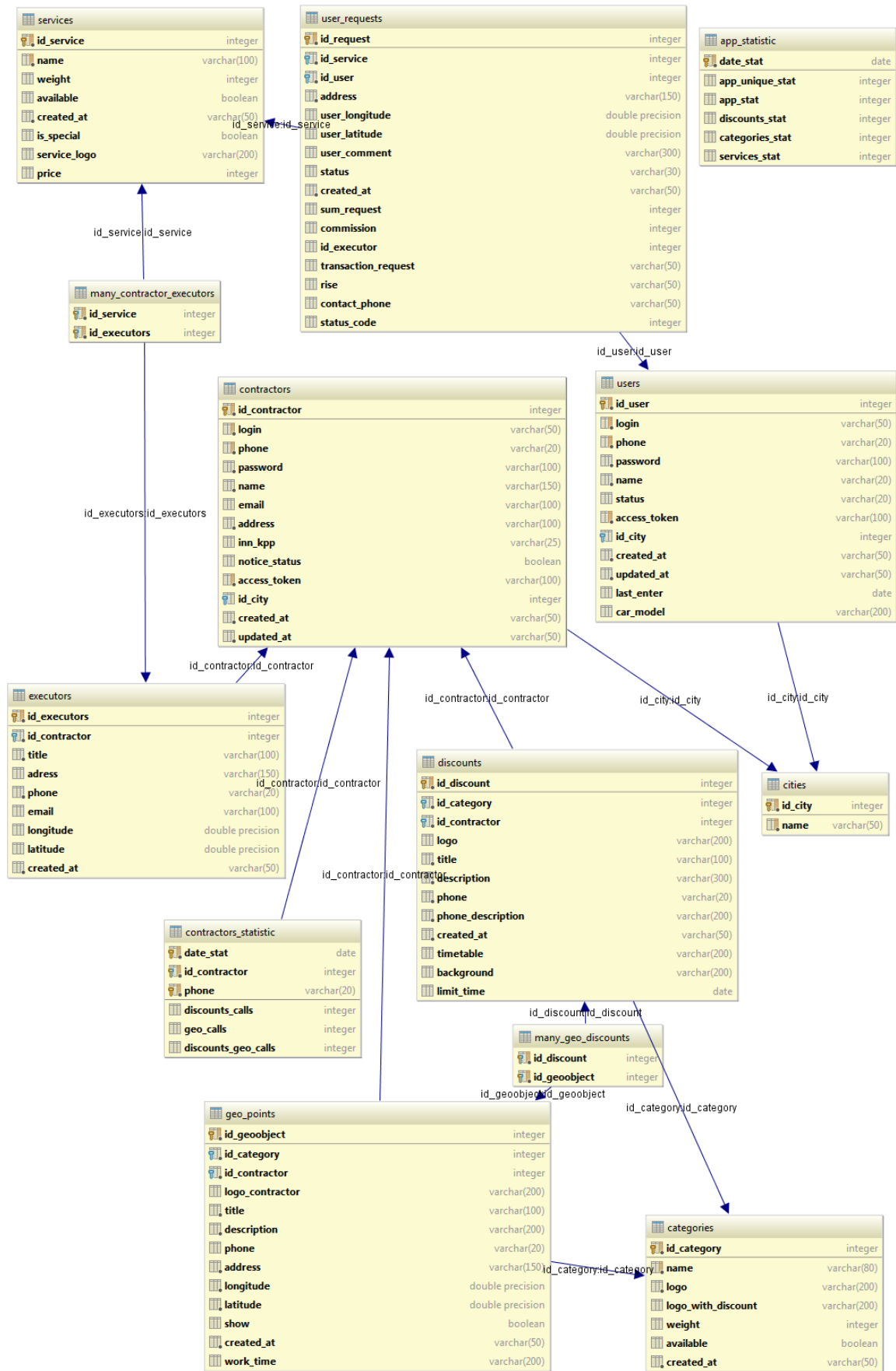


Рисунок 1 – Логическая модель

разработанная компанией JetBrains, а в качестве среды разработки для Android приложения была выбрана официальная IDE для этих целей — «Android Studio».

2.3.1 IntelliJ IDEA

IntelliJ IDEA — интегрированная среда разработки программного обеспечения, она действительно является мощнейшим редактором исходного кода. В ней собраны не только классические подсветки синтаксиса, расстановка табуляции и несколько буферов обмена, но и быстрые переходы по коду, автоматическое добавление нужных импортов, а также оптимизация импортов с удалением ненужных, встроенное отображение документации и типов, подсказка по аргументам, превосходная система рефакторинга, быстрое указание возвращаемого типа у метода, различные подсказки на улучшение качества кода.

2.3.2 Android Studio

Android Studio — это интегрированная среда разработки для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O. Основана на программном обеспечении IntelliJ IDEA от компании JetBrains и является официальным средством разработки Android приложений.

2.4 Необходимые технологии

Было решено, что Spring Framework станет сердцем разрабатываемого сервиса помощи автолюбителям. С помощью этого фреймворка будет разработана вся backend часть проекта, а именно RESTful web-сервис. Будет реализована обработка HTTP-запросов от пользовательских клиентов и от web админки бизнес пользователя, а также доступ к базе данных и защита от несанкционированного доступа.

В качестве пользовательских клиентов будет выступать мобильные приложения на базе операционной системы Android. Мобильные устройства стали неотъемлемой частью жизни людей, а доля Android-смартфонов составляет 85% по итогам 2016 года, что является несомненным доводом, чтобы охватить большую аудиторию.

Административная панель для бизнес пользователя, которая будет включать себя возможность полностью управлять сущностями сервиса, а также ве-

сти некоторую статистику, будет реализована в качестве одностраничного web-приложения с помощью современного javascript фреймворка — «AngularJS».

2.4.1 Maven

Maven был выбран в качестве системы автоматической сборки проекта серверной и панели администратора. Это инструмент для сборки Java проекта: компиляции, создания jar, создания дистрибутива программы, генерации документации. Простые проекты можно собрать в командной строке. Если собирать большие проекты с командной строки, то команда для сборки будет очень длинной, поэтому её иногда записывают в bat/sh скрипт. Но такие скрипты зависят от платформы. Для того чтобы избавиться от этой зависимости и упростить написание скрипта используют инструменты для сборки проекта.

2.4.2 Spring Framework

Spring Framework обеспечивает комплексную модель разработки и конфигурации для современных бизнес-приложений на Java — на любых платформах. Ключевой элемент Spring — поддержка инфраструктуры на уровне приложения: основное внимание уделяется «водопроводу» бизнес-приложений, поэтому разработчики могут сосредоточиться на бизнес-логике без лишних настроек в зависимости от среды исполнения.

2.4.3 Spring MVC

Spring MVC модуль отлично подходит для реализации RESTful web-сервиса.

В основе модуля Spring MVC лежит стандарт MVC, который состоит из трех компонентов:

- Models — объекты на которую возлагается бизнес логика;
- Views — представление данных (html, jsp, json, xml и т.д.);
- Controller — своего рода мост для отображения на views того, что происходит в models.

2.4.4 Spring Security

Spring Security — это модуль, который сфокусирован на обеспечение как аутентификации, так и авторизации в Java-приложениях. Как и все Spring проекты, настоящая сила Spring Security в том, что он может быть легко дополнен нужным функционалом.

2.4.5 Spring JdbcTemplate

Spring предоставляет шаблонный класс `JdbcTemplate`, который упрощает работу с SQL и JDBC реляционных СУБД. Большая часть JDBC кода состоит из получения ресурсов, управления соединением, обработки исключений и вообще проверки ошибок, ни как не связанных с тем, для чего предназначен код. `JdbcTemplate` берет все это на себя. Благодаря этому при работе с БД в Spring Framework, нам необходимо только определить параметры соединения с БД и прописать SQL-запрос. Программисту остается только сосредоточиться на решении поставленной задачи.

2.4.6 Spring Boot

Spring Boot — инструмент для упрощения процесса конфигурации Spring приложений. Это не является средством автоматической генерации кода, а представляет собой плагин для системы автоматизации сборки проектов таких как Maven или Gradle.

Главное преимущество Spring Boot — конфигурирование ресурсов исходя из содержания classpath. Например, если `pom.xml` файл вашего Maven проекта содержит веб-зависимости, то вы получите сконфигурированный по умолчанию Spring MVC. Если вы создаете веб-приложение, но не указываете ничего дополнительно, плагин сконфигурирует `view resolver` для системы шаблонов Thymeleaf, а `java` бины будут по умолчанию сериализоваться в `json` формат при построении REST сервиса.

2.4.7 Angular JS

Было решено, что frontend часть панели администратора для бизнес пользователя будет реализована через популярный javascript-фреймворк — AngularJS. Он предназначен для разработки одностраничных приложений. Его цель — расширение браузерных приложений на основе MVC шаблона, а также упрощение тестирования и разработки.

2.4.8 Bootstrap

Верстать веб-страницы является довольно сложным и монотонным процессом. Поэтому каждый веб-разработчик и верстальщик рано или поздно задумывается о том, как ему упростить и ускорить процесс верстки сайта. В

связи с этим, он прибегает к помощи css-фреймворков. Самый популярный из них — bootstrap.

2.4.9 Android

Смартфоны и планшеты на Android все чаще встречаются в сумках и карманах людей, и программирование под Android также становится все популярнее. Это отличная платформа для разработки — API прекрасно документировано и довольно просто в использовании. Поэтому данная платформа была выбрана в качестве пользовательского клиента.

2.4.10 Apache HttpClient

Протокол передачи гипертекста (HTTP) — это, пожалуй, самый важный протокол, используемый в Интернете на сегодняшний день. Веб-сервисы, сетевые устройства и рост сетевых вычислений продолжают расширять роль HTTP-протокола вне пользовательских веб-браузеров, одновременно увеличивая количество приложений, требующих поддержки HTTP.

2.4.11 Система автоматической сборки Gradle

Gradle — система автоматической сборки, построенная на принципах Apache Maven, но предоставляющая описание сценариев на языке Groovy вместо традиционной XML-образной формы представления конфигурации проекта.

2.4.12 Google Maps API

С помощью Google Maps Android API можно добавлять в свое приложение карты на основе данных Google Maps, что и требуется для разработки мобильного приложения в соответствии с техническим заданием.

ЗАКЛЮЧЕНИЕ

В ходе дипломной работы был разработан полноценный web-сервис с возможностью будущего коммерческого использования.

Для достижения результата были использованы следующие книги, электронные источники и технологии:

1. Для фундаментального понимания веб-сервисной архитектуры была прочитана книга М. Калина «Java Web Services: Up and Running» [1], а также книга Т.С. Машнина «Web-сервисы Java» [2].
2. Spring Framework и его компоненты для написания серверной части, используя официальную документацию с ресурса [4], а также книгу Г. Мака «Spring Recipes: A Problem-Solution Approach» [5].
3. При проектировании базы данных использовалась официальная документация PostgreSQL [6], информация из учебника Н. Стружина «Базы данных: Проектирование» [3], а также информация статей «Логическая модель предметной области» [7] и «Проектирование баз данных. Нормализация.» [8].
4. При работе с функционалом автоматического сборщика проектов Maven, была рассмотрена информация из официального электронного ресурса сборщика [9], содержащий всю необходимую информацию, помимо этого стоит отметить статью «Maven for building Java applications» [10] за отличный материал и практические примеры.
5. Во время работы над мобильным приложением книга Голощапова А. Л. «Google Android. Создание приложений для смартфонов и планшетных ПК» [11] и статья «Android Tutorial» [12] с практическими примерами были крайне полезны во время разработки, и в течении всего времени проходил процесс изучения официальной документации по работе с android API [13].
6. Для использования Google Maps Android API, Google Analytics, Firebase Cloud Messaging в мобильном приложении, были изучены соответствующие официальные веб-страницы для разработчиков:
 - «developers.google.com/maps/documentation/android-api/» [14];
 - «[developers.google.com/analytics/devguides/collection/android/v4 /](https://developers.google.com/analytics/devguides/collection/android/v4/)» [15];
 - «firebase.google.com/docs/cloud-messaging/» [16].
7. Во время разработки HTML макета панели администратора, информа-

ция с электронного ресурса «www.w3schools.com/bootstrap» [17] была незаменима.

8. При изучении AngularJS — основным источником информации стала его официальная документация [18] и книга П. Козловского «Разработка веб-приложений с использованием AngularJS» [19].

В результате выполнения работы были отработаны навыки:

- программирование на языке Java;
- работа со сторонними библиотеками и фреймворками;
- построение архитектуры проекта согласно техническому заданию;
- разработка web и Android приложения;
- построение логической и физической модели базы данных;
- подбор необходимые инструменты для разработки веб-сервисов;
- проектирование и разработка простой и масштабируемую архитектуры веб-сервиса.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Kalin, M.* Java Web Services: Up and Running, 2nd Edition / M. Kalin. — O'Reilly Media, 2013. — P. 360.
- 2 *Машнин, Т. С.* Web-сервисы Java / Т. С. Машнин. — БХВ-Петербург, 2012. — P. 560.
- 3 *Стружкин, Н. П.* Базы данных: Проектирование / Н. П. Стружкин. — Paul Manning, 2017. — P. 478.
- 4 Spring documentation. [Электронный ресурс]. — URL: <https://spring.io/docs> (Дата обращения 6.05.2017). Загл. с экр. Яз. англ.
- 5 *Mak, G.* Spring Recipes: A Problem-Solution Approach / G. Mak. — Paul Manning, 2010. — P. 1058.
- 6 PostgreSQL: Documentation. [Электронный ресурс]. — URL: <https://www.postgresql.org/docs/> (Дата обращения 7.05.2017). Загл. с экр. Яз. англ.
- 7 Логическая модель предметной области. [Электронный ресурс]. — URL: <http://analyst.by/diagrams/logicheskaya-model-predmetnoy-oblasti> (Дата обращения 24.05.2017). Загл. с экр. Яз. рус.
- 8 Проектирование баз данных. Нормализация. [Электронный ресурс]. — URL: <http://club.shelek.ru/viewart.php?id=311> (Дата обращения 27.05.2017). Загл. с экр. Яз. рус.
- 9 Apache Maven Project. [Электронный ресурс]. — URL: <https://maven.apache.org/index.html> (Дата обращения 6.05.2017). Загл. с экр. Яз. англ.
- 10 Maven for building Java applications. [Электронный ресурс]. — URL: <http://www.vogella.com/tutorials/ApacheMaven/article.html> (Дата обращения 7.05.2017). Загл. с экр. Яз. англ.
- 11 *Голощанов, А. Л.* Google Android. Создание приложений для смартфонов и планшетных ПК. / А. Л. Голощанов. — БХВ-ПЕТЕРБУР, 2013. — P. 830.
- 12 Android Tutorial. [Электронный ресурс]. — URL: <https://www.tutorialspoint.com/android/> (Дата обращения 17.05.2017). Загл. с экр. Яз. англ.

- 13 Introduction to Android. Android Developers. [Электронный ресурс].— URL: <https://developer.android.com/guide/index.html> (Дата обращения 15.05.2017). Загл. с экр. Яз. англ.
- 14 Google Maps Android API. [Электронный ресурс].— URL: <https://developers.google.com/maps/documentation/android-api/> (Дата обращения 15.05.2017). Загл. с экр. Яз. англ.
- 15 Google Analytics for Android. [Электронный ресурс].— URL: <https://developers.google.com/analytics/devguides/collection/android/v4/> (Дата обращения 25.05.2017). Загл. с экр. Яз. англ.
- 16 Firebase Cloud Messaging. [Электронный ресурс].— URL: <https://firebase.google.com/docs/cloud-messaging/> (Дата обращения 25.05.2017). Загл. с экр. Яз. англ.
- 17 Bootstrap 3 Tutorial. [Электронный ресурс].— URL: <http://www.w3schools.com/bootstrap/> (Дата обращения 17.05.2017). Загл. с экр. Яз. англ.
- 18 AngularJS API Docs. [Электронный ресурс].— URL: <https://docs.angularjs.org/api> (Дата обращения 13.05.2017). Загл. с экр. Яз. англ.
- 19 *Козловский, П.* Разработка веб-приложений с использованием AngularJS / П. Козловский. — ДМК Пресс, 2014. — Р. 394.

10.06.2017

