

Министерство образования и науки Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**Разработка автоматизированной системы «Свободная очередь  
(FreeQ)». Реализация клиентских сервисов, интерфейсов  
администраторов и уведомлений пользователей.**

**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студентки 4 курса 441 группы  
направления 02.03.03 Математическое обеспечение и администрирование  
информационных систем факультета компьютерных наук и  
информационных технологий  
Макарова Алексея Александровича

Научный руководитель

Руководитель к.ф.-м.н., доцент

А.Г. Федорова

Консультант

АО «СберТех» главный ИТ-инженер

Соломина Ю. Ю.

Зав. кафедрой

к.ф.-м.н.

М.В. Огнева

Саратов 2018

## ВВЕДЕНИЕ

**Актуальность темы.** Проблема очередей сейчас особенно актуальна. Очереди являются неотъемлемой частью нашей жизни и встречаются буквально на каждом шагу: в поликлиниках, банках и даже в учебных заведениях при подаче документов. И по большей части они представляют собой неудобную структуру, в которой часто возникают проблемы, связанные со временем. Что делать при опоздании? Что делать, если перед тобой есть опаздывающие люди? Такие вопросы наиболее часто возникают у людей, которые находятся в плохо структурированной очереди.

Автоматизация очереди является естественным решением и призвана решить основные проблемы использования очереди. Однако классическая очередь совершенно не годится для автоматизации, поэтому за основу была взята идея «Свободной очереди».

«Свободная очередь» — это очередь, в которой ожидание обслуживания сводится к минимуму за счет механизма динамической балансировки. Пользователю такой системы нет смысла переживать, если он опаздывает.

**Цель бакалаврской работы** — разработка интерфейсов администраторов, клиентских сервисов и создание системы уведомлений для пользователей.

Поставленная цель определила следующие задачи:

1. Изучить способы разработки приложений.
2. Выбрать наиболее подходящий для разработки язык.
3. Выбрать инструменты и технологии для создания приложения.
4. Изучить способы развертывания приложений в облаке.
5. Изучить различные технологии уведомлений.
6. Разработать приложение для администраторов позволяющее манипулировать очередями и билетами, а также добавлять и удалять новые сервисы и очереди.

7. Разработать серверное приложение для пользователей позволяющее регистрироваться и авторизоваться в приложении, подключаться к очереди и следить за своим расположением в ней.

**Методологические основы** для разработки современных приложений представлены в работах Д. Самсонов, С. Фомин [1], С. Макконнелл [7], Г. Шилдт [8], Д. Влсидис, Р. Джонсон, Р. Хелм, Э. Гамма [11].

**Практическая значимость бакалаврской работы.** В ходе этой работы были разработаны два приложения из большого проекта по построению системы для управления автоматизированными очередями, которые будут использовать другие разработчики в дальнейшем.

Как только все части приложения будут готовы, данное приложение будет использоваться для разрешения сложных ситуаций в местах, где может использоваться структура свободной очереди, позволяющая сделать взаимодействие с очередью наиболее удобным для клиента.

**Структура и объём работы.** Бакалаврская работа состоит из введения, 3 разделов, заключения, списка использованных источников и 1 приложения. Общий объём работы – 66 страниц, из них 59 страниц – основное содержание, включая 22 рисунка, цифровой носитель в качестве приложения, список использованных источников информации – 20 наименований.

## КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

### **Первый раздел «Этапы и технологии разработки web-приложений»**

посвящен общим сведениям о разработке web-приложений. В данной главе рассматриваются этапы и общие принципы разработки приложений такого типа. Подробно описываются шаги, которые нужно выполнить для достижения успеха в разработке данного приложения.

Также описываются различные методологии разработки, такие как каскадная, гибкая или итеративная модели [1, 2]. Были рассмотрены их плюсы и минусы, а также выбрана наиболее подходящая методология для разработки проекта «Свободная очередь» – Каскадная модель.

В следующем параграфе рассматриваются различные системы контроля версий (VCS), которые могут быть применены для создания комфортных условий при разработке. Были рассмотрены такие VCS, как Git, SVN [3, 4, 5], Mercury. В итоге для разработки, наиболее подходящей оказалась Git, под использованием крупного графического репозитория Gitlab.

Также были изучены принципы написания технической документации. Описано, зачем нужно писать документацию во время разработки, а также какие плюсы из этого можно извлечь.

В заключении раздела были изучены различные облачные сервисы, позволяющие развернуть приложение. Рассматривались два различных по функциям сервиса – это Heroku и Open Shift. Но в конечном итоге был выбран первый, так как он позволял бесплатно разворачивать несколько приложений, что было крайней полезно для проекта, так как он имел несколько разных больших модулей, каждый из которых и был отдельным приложением.

**Второй раздел «Языки и средства создания web и мобильных приложений, использованные при создании автоматизированной системы Свободная очередь (FreeQ)» посвящен описанию различных средств и инструментов, которые были использованы при разработке приложения.**

В первую очередь был подробно описан язык Java, на котором велась разработка приложения. Были описаны его история и основные особенности, такие как виртуальная машина, сборщик мусора, компиляция в реальном времени, рассмотрены плюсы и минусы, а также принципы разработки, которые он предлагает, то есть концепция объектно-ориентированного программирования. В заключении параграфа были рассмотрены различные среды разработки, которые позволяют сделать программирование на языке Java более быстрым и удобным [6, 7, 8, 9].

Далее был рассмотрен основной фреймворк, на котором велась разработка приложения – Spring. Изучены основные возможности, которые предоставляет данный фреймворк:

1. Контейнер зависимостей.
2. Data.
3. Web.

Были рассмотрены различные паттерны, которые предлагает Spring для своего использования, например, Singleton и Prototype. Также был рассмотрен основной принцип фреймворка – Convention over configuration (Конвенция превыше конфигурации). Помимо всех преимуществ, были рассмотрены недостатки использования такого мощного инструмента для разработки, основным из которых является конечно же сложность [10, 11, 12].

Была рассмотрена отдельная часть фреймворка под названием Spring Roo, которая используется для быстрого создания каркаса приложения, на основе объектной модели. Но данный фреймворк, все ещё находится на

стадии разработки, поэтому в процессе использования возникало множество ошибок, на решение которых уходило много времени [13].

Также были рассмотрены различные вспомогательные библиотеки, облегчающие написание кода в тех или иных случаях. Google Guava – библиотека, включая в себя множество стандартных функциональных классов, расширяющих утилитные возможности языка. Lombok – позволяющая избавиться от устаревшего boilerplate кода. Org.Json – одна из старейших библиотек для манипуляции JSON документами, имеющая самый простой API среди всех аналогов [14, 15, 16].

В параграфе 2.5 была рассмотрена самая популярная система сборки исходного кода Maven. Были изучены различные аналоги и принципы работы данного инструмента, такие как Ant и Gradle, а также задачи, которые он решает [17, 18].

Последние технологии, которые были рассмотрены – это различные способы уведомления пользователей. Описаны два совершенно разных способа: Google Cloud Message и SMS-Gateway, первый из которых является бесплатным, но имеет различные недостатки, например, отсутствие гарантии доставки и невозможность привязки к номеру, второй не имеет таких недостатков, но не является бесплатным. В дальнейшем в проекте были использованы оба способа, но для разных целей.

После рассмотрения различных инструментов были составлены требования к приложению «Свободная очередь». Описаны действия, которые может делать каждый из типов пользователей, которые присутствуют в системе: поставщик услуг, клиент и администратор. В заключении раздела была описана схема взаимодействия всех модулей в системе «Свободная очередь».

**Третий раздел «Реализация клиентских сервисов, интерфейсов администраторов и уведомлений пользователей для автоматизированной системы Свободная очередь (FreeQ).»** посвящен непосредственно разработке компонент системы.

Первым шагом была описана архитектура приложения, включающая в себя взаимодействие различных функциональных модулей между собой. Данная часть разработки будет включать в себя написание двух приложений: для администраторов и серверной части для пользователей.

Пользовательское приложение будет делиться на несколько функциональных частей, таких как:

1. Аутентификация.
2. Управление данными пользователя.
3. Поиск.
4. Интеграция.
5. Управление очередями.

Далее рассматривается составление объектной модели, которая будет использоваться для создания таблиц в базе данных и POJO классов в самих приложениях.

Отдельный параграф посвящен рассмотрению разработки с помощью фреймворка Spring Roo и построению приложения для администраторов. Был описан непосредственно процесс работы с консолью, а также разобраны несколько команд в качестве примера.

Далее более подробно с примерами кода описывается создание серверной части для пользователей. В параграфе 3.4 описывается создание модуля для аутентификации, а также основные принципы, «привязывания» пользователей к приложению. Аутентификация основана на номере телефона. Процесс включает в себя отправку секретного кода через SMS, на основе которого выполняется идентификация пользователя в приложении.

Следующим шагом рассмотрены различные способы поиска услуг в приложении. Было разработано API для полнотекстового поиска по различным мета-тегам, которые настраивает администратор. Кроме этого, была реализована система поиска по координатам на карте.

Далее описаны процессы интеграции со сторонними сервисами: Google Cloud Message и SMS-Gateway. В основном эти процессы включают в себя аутентификацию в соответствующем сервисе и последующую отправку запроса на определенный HTTP-endpoint.

Последним шагом в разработке было создание модуля для управления очередями, который включает в себя разные функции по определению времени активности очереди, по присоединению к активной очереди, по мониторингу своего положения в очереди и обработку ситуации опоздания клиента.

В заключительном параграфе был описан процесс развертки приложения в облачном сервисе Heroku, а также команды, которые необходимо выполнить для того, чтобы развернуть свое приложение.



## ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы были созданы два модуля из общей системы управления очередями, отвечающие за работу с пользователями и администраторами. Сервис по управлению пользователями представляет собой сервер, предоставляющий REST-API для взаимодействия с системой. Сервис для администраторов представляет собой отдельное приложение с серверной и клиентской частью, созданными при помощи технологии генерации кода Spring Roo.

Для создания приложения были изучены инструменты, позволяющие реализовать различные его модули. В качестве каркаса приложения использовался Spring Framework, предоставляющий функционал по управлению объектами классов в приложении. Для работы с базой данных использовалась технология Java Persistence API, позволяющая упростить взаимодействие с базой данных из Java кода, а также создать взаимное соответствие Java-классов и таблиц в базе данных. Для реализации шаблона MVC использовался один из модулей Spring Framework под названием Spring Web, позволяющий создавать контроллеры, запускающие определенную логику при обращении к URL ресурсу на сервере.

Кроме того, были изучены два вида уведомлений пользователей: push и SMS. Была проведена интеграция со сторонними сервисами, предоставляющими функционал по рассылке уведомлений пользователям. В качестве сервиса рассылки push-уведомлений использовался Google Cloud Message, позволяющий бесплатно отправлять короткие сообщения в клиентские приложения пользователей. Для SMS-рассылки использовалась технология SMS-Gateway, позволяющая рассылать уведомления при помощи протокола http.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Д. Самсонов, С. Фомин «Структура и методология разработки автоматизированных информационных систем», изд. МИСиС 2005г. — 239 стр.
2. С. С. Куликов «Тестирование программного обеспечения», 2-е издание — 296 стр.
3. С. Чакоп, Б. Штрауб «Git для профессионального программиста» изд. Питер, 2017 год — 496 стр.
4. Б. Коллинс-Сассман «Управление версиями в Subversion» 2007 г. — 146 стр.
5. Б. О'шалливан «Mercurial: Полное руководство» 2009г. — 193 стр.
6. Б. Эккель «Философия Java» (ориг. название «Thinking in Java»), изд. Питер 1998г. — 1168 стр.
7. С. Макконнелл «Совершенный код» изд. «БХВ» 2017 г. — 867 стр.
8. Г. Шилдт «Java 8. Полное руководство 9-е издание» изд. «Вильямс» 2015 г. — 1377 стр.
9. Х. Мессенбок «Плюсы и минусы объектно-ориентированного программирования» изд. ИнфоАрт 1995г. — 278 стр.
10. *Inversion of Control Containers and the Dependency Injection pattern* [Электронный ресурс]. — URL: <https://www.martinfowler.com/articles/injection.html>  
(Дата обращения 10.04.2018) Загл. с экр. Яз. англ.
11. Д. Влсидис, Р. Джонсон, Р. Хелм, Э. Гамма «Приёмы объектно-ориентированного проектирования. Паттерны проектирования» изд. «Питер» 2014г. — 395 стр.
12. *Spring Framework Reference Documentation* [Электронный ресурс]. — URL: <https://docs.spring.io/spring/docs/4.3.17.RELEASE/spring-framework-reference/htmlsingle/> (Дата обращения 15.03.2018) Загл. с экр. Яз. англ.

13. *Spring Roo Reference Documentation* [Электронный ресурс]. — URL: <https://docs.spring.io/spring-roo/docs/current/reference/html/>  
(Дата обращения 20.04.2018) Загл. с экр. Яз. англ.
14. *Reducing Boilerplate Code with Project Lombok* [Электронный ресурс]. — URL: <http://jnb.ociweb.com/jnb/jnbJan2010.html>  
(Дата обращения 02.04.2018) Загл. с экр. Яз. англ.
15. *Google Core Libraries for Java* [Электронный ресурс]. — URL: <https://opensource.google.com/projects/guava>  
(Дата обращения 05.03.2018) Загл. с экр. Яз. англ.
16. *Introducing JSON* [Электронный ресурс]. — URL: <https://www.json.org/>  
(Дата обращения 05.02.2018) Загл. с экр. Яз. англ.
17. *Welcome to Apache Maven* [Электронный ресурс]. — URL: <https://maven.apache.org/> (Дата обращения 05.01.2018) Загл. с экр. Яз. англ.
18. В. Джейсон «Maven: The Definitive Guide» изд. «O'Reilly Media» 2009 г. — 470 стр.
19. *MESSAGES REST V2* [Электронный ресурс]. — URL: [https://prostor-sms.ru/smsapi/rest\\_messages\\_api.prostor-sms.ru.pdf](https://prostor-sms.ru/smsapi/rest_messages_api.prostor-sms.ru.pdf)  
(Дата обращения 05.05.2018) Загл. с экр. Яз. Англ.
20. *Google Cloud Messaging Overview* [Электронный ресурс]. — URL: <https://developers.google.com/cloud-messaging/gcm>  
(Дата обращения 15.04.2018) Загл. с экр. Яз. Англ.