

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**

**Тема: Описание предполагаемого способа решения**

Студент \_\_\_\_\_

гр. 5304 Фомченко О.А.

Руководитель \_\_\_\_\_

Лисс А.А.

Санкт-Петербург 2020

## Содержание:

АРХИТЕКТУРА ПРИЛОЖЕНИЯ.....	3
Описание системы:.....	3
Описание архитектуры:.....	3
ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ.....	4
Описание: .....	4
Предназначение в приложении:.....	4
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	6

## АРХИТЕКТУРА ПРИЛОЖЕНИЯ

### Описание системы:

Разрабатываемое приложение представляет собой веб-приложения, написанное на языке программирования Java. Для осуществления взаимодействия между front-end и back-end используется REST api и web-socket (например для мессенджера), которые передают данные в формате JSON. Для сериализации и десериализации JSON объектов будет использоваться библиотека Jackson.

Все данные хранятся в базе данных PostgreSQL. Взаимодействие между базой данных и приложением осуществляется при помощи Hibernate и Spring Data через DAO (Data Access Object) сервисный слой.

Приложение использует Spring IoC (Inversion of Control) контейнер для управления жизненного цикла объектов, что позволяет эффективно использовать ресурсы сервера.

Для обеспечения безопасности данных приложения использует Spring Security, позволяющую создавать систему ролей и осуществлять надежную авторизацию и регистрацию.

### Описание архитектуры:

Разрабатываемое приложение имеет микросервисную архитектуру, что позволяет улучшить гибкость и масштабируемость системы.

Приложение состоит из следующих компонентов:

- Мессенджер – сервис, отвечающий за систему коммуникаций между пользователями приложения.
- Трекер задач – сервис, отвечающий за управление, создание и распределение целей и задач команды.
- Графический интерфейс – сервис, отвечающий за отображение информации в браузере.
- Главный модуль – сервис, отвечающий за управление зависимостями между командами и пользователями.

## ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ

### Описание:

- Spring – архитектурный фреймворк с подключаемыми модулями, используемый для разработки веб-приложений на Java.
- Gradle – система автоматической сборки проекта.
- Git – распределенная система контроля версий.
- Docker - программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.
- Jenkins – программная система, предназначенная для осуществления процесса непрерывной интеграции.
- PostgreSQL – объектно реляционная система управления базами данных.
- Hibernate – система решения задач объектно реляционного отображения для Java.
- Kubernetes – платформа для управления контейнеризированными рабочими нагрузками и сервисами.
- Apache Tomcat – контейнер веб-приложений.

### Предназначение в приложении:

- Spring – используется в качестве базового фреймворка, являющегося основой приложения. Так же в проекте используются некоторые модули, такие как Spring Data, Spring Security и т.д.
- Gradle – используется в качестве сборщика проекта, для удобного подключения зависимостей и создания job-ов.
- Git – используется для хранения кодовой базы.
- Docker – используется для удобного развертывания приложения и его окружения.
- Jenkins – используется в качестве системы непрерывной интеграции.
- PostgreSQL – базовая система управления базами данных.
- Hibernate – используется для осуществления эффективного взаимодействия между базой данных и приложением.
- Kubernetes – используется для эффективного управления сервисами приложения и их окружения.

- Apache Tomcat – используется в качестве самостоятельного веб-сервера, на котором работает приложение.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Микросервисная архитектура \ блог компании Райффайзен банка  
<https://habr.com/ru/company/raiffeisenbank/blog/346380/>
2. Kubernetes / Документация  
<https://kubernetes.io/ru/docs/concepts/overview/what-is-kubernetes/>
3. Spring with REST api / Документация  
<https://spring.io/guides/tutorials/rest/>
4. Spring with microservices / Документация  
<https://spring.io/microservices>
5. Spring boot / Быстрый старт  
<https://spring.io/projects/spring-boot>