

Національний технічний університет України  
«Київський політехнічний інститут»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

**Розрахунково-графічна робота**  
по курсу  
«ІНТЕГРАЦІЙНІ ПРОГРАМНІ СИСТЕМИ»

Виконали: студенти 4 курсу  
ФІОТ гр. ІО-43  
Команда helloworld! в складі:  
Захарчук Д. В.  
Талєб Я. Б.  
Смоляр О. В.  
Богуславська К. С.

Київ 2017

## Back-end

Github: <https://github.com/yktaleb/ProblemOff>

Domain: <https://problemoff.herokuapp.com/>

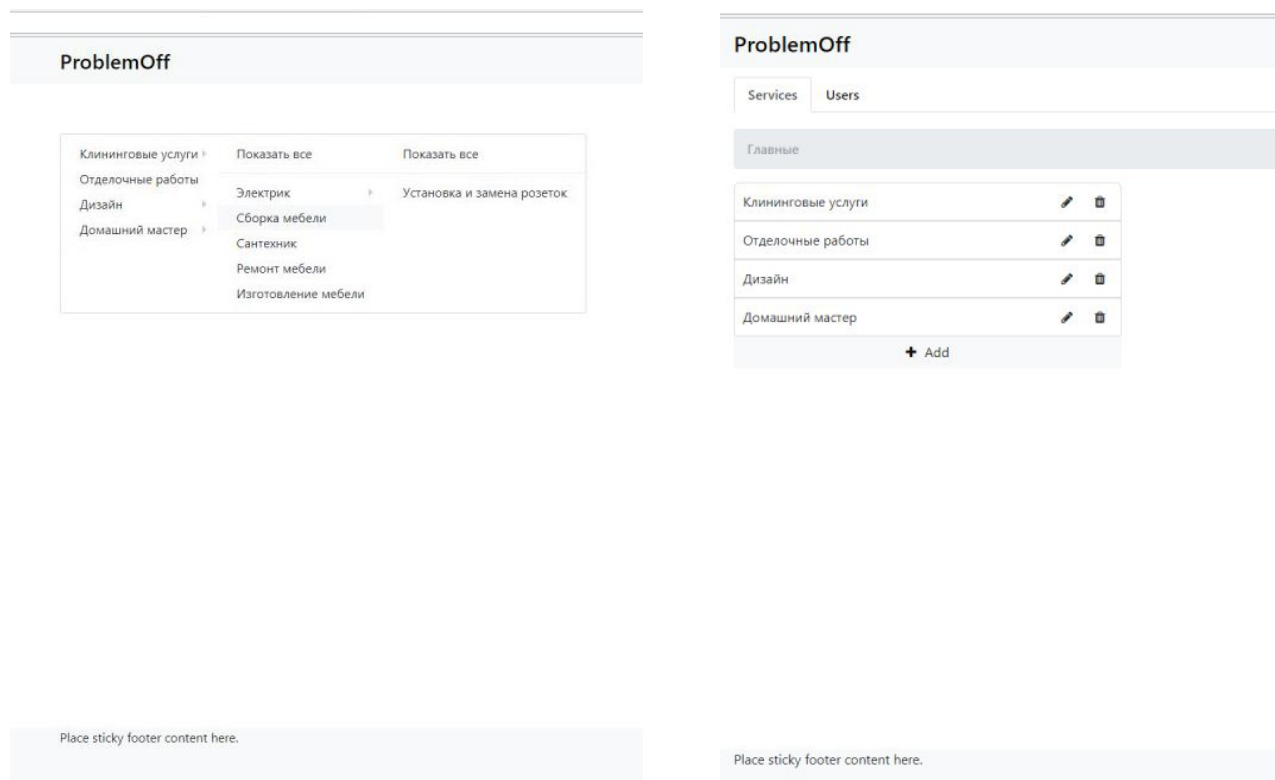
## Front-end

Github: <https://github.com/dzakhark/problem-off>

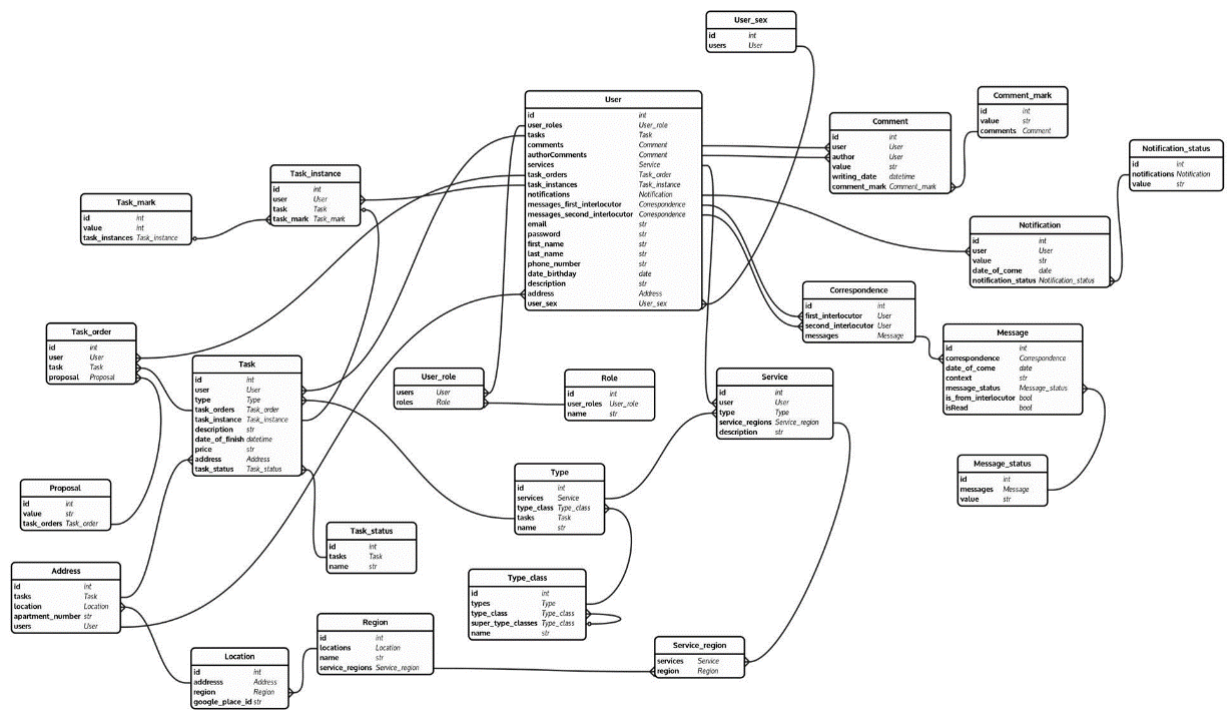
Domain: <https://problem-off.firebaseio.com/>

### 1. Опис проекту

Проект, розроблений у ході виконання циклу лабораторних робіт, являє собою систему для найму робітників на довго та короткотривалі роботи. З боку ж заробітчан, ProblemOff надає додаткові шляхи пошуку роботи згідно категорії вмінь. Проект орієнтовано на короткотривалі побутові (зазвичай ремонтні) роботи. Прибутковість проекту, як і в інших фріланс-біржах, базується на оплачуваному посередництві.



Скріншоти роботи сервісу



Було спроектовано ERD, зображену на малюнку.

Серверна частина сервісу розроблена за допомогою засобів Java. В основі архітектури лежить паттерн MVC, який реалізовано за допомогою фреймворку Spring MVC. Використовується база даних PostgreSQL, яка знаходиться на віддаленому сервері.

Для зменшення кількості зв'язків між серверною та фронт-енд частиною використовується Hypermedia. Для більш комфортної роботи з Hypermedia застосовано модуль Spring HATEOAS.

Для гарного масштабування веб-сервісу в клауд-системі розроблено token based authentication з допомогою Spring Security та JWT.

Інтерфейс реалізовано за допомогою компонентного підходу фреймворку Angular4 та Bootstrap4.

## 2. Релізація атоматичної збірки.

Для збірки бек-енду використовується Maven, котрий, зв'язуючись з центральним репозиторієм, завантажує необхідні для функціонування проекту залежності.

Для збірки використовується плагін `spring-boot-maven-plugin`, а також плагін для статичного аналізу коду.

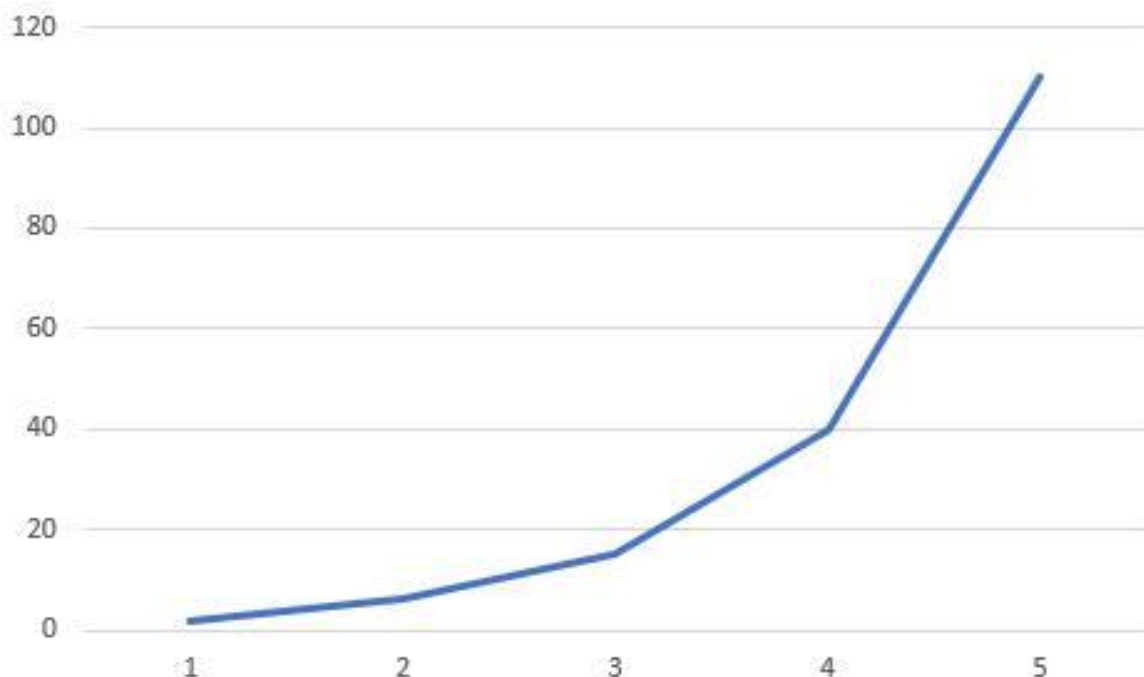
Збірка фронт-енд частини реалізована за допомогою засобів `Angular CLI`, який керується файлом `package.json`, що в свою чергу містить всі залежності для розробки та роботи інтерфейсу.

### 3. Сервер безперервної інтеграції. Travis CI

У проєкті використовується система безперервної інтеграції `Travis CI`, що пов'язана з `GitHub` репозиторієм. `Travis CI` реагує на внесення змін до пов'язаної з нею репозиторієм, щоразу виконуючи збірку проєкту з внесеними змінами, запускаючи тести та, у випадку їх успішного проходження, деплоєм на `cloud system`. Такими системами для бек та фронт-енд частини є відповідно `Heroku` та `Firebase`.

З'язок між серверною та інтерейсною частинами реалізовано за допомогою `Арі`, що надає сервер. Для подолання проблеми блокування міждоменого запиту використовується специфікація технології браузерів `CORS`, що є сучасною альтернативою `JSONP`.

### 4. Експоненціальна витримка



Для вирішення задачі оптимізації кількості та частоти надсилання повторних запитів на сервер, у випадку невдалого попереднього запиту, створено стратегію щодо збільшення часового інтервалу між повторними спробами встановити зв'язок. Графік залежності зміни інтервалу між запитами до серверу від часу, що минув з першого невдалого запиту зображено на малюнку.