Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

*Факультет Программной инженерии и компьютерной техники*

**Лабораторная работа 4**

Вариант № 1948

**Студент**: Фролов К. Д.

**Группа:** P3206

**Преподаватель**: Письмак А. Е.

г. Санкт-Петербург

2023г

# Оглавление

[Оглавление 2](#_Toc152761821)

[Задание 2](#_Toc152761822)

[Программа 3](#_Toc152761823)

[Заключение 4](#_Toc152761824)

[Ссылки 4](#_Toc152761825)

# Задание

Переписать приложение из [предыдущей лабораторной работы](https://se.ifmo.ru/en/courses/web#lab3) с использованием следующих технологий:

* Уровень back-end должен быть основан на Spring.
* Уровень front-end должен быть построен на [Angular 2+](https://angular.io/" \t "_blank) с использованием набора компонентов [PrimeNG](https://www.primefaces.org/primeng/" \l "/" \t "_blank)
* Взаимодействие между уровнями back-end и front-end должно быть организовано посредством REST API.

Приложение по-прежнему должно включать в себя 2 страницы - стартовую и основную страницу приложения. Обе страницы приложения должны быть адаптированы для отображения в 3 режимах:

* "Десктопный" - для устройств, ширина экрана которых равна или превышает 1184 пикселей.
* "Планшетный" - для устройств, ширина экрана которых равна или превышает 814, но меньше 1184 пикселей.
* "Мобильный"- для устройств, ширина экрана которых меньше 814 пикселей.

**Стартовая страница должна содержать следующие элементы:**

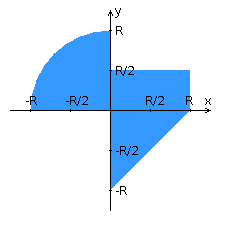
* "Шапку", содержащую ФИО студента, номер группы и номер варианта.
* Форму для ввода логина и пароля. Информация о зарегистрированных в системе пользователях должна храниться в отдельной таблице БД (пароль должен храниться в виде хэш-суммы). Доступ неавторизованных пользователей к основной странице приложения должен быть запрещён.

**Основная страница приложения должна содержать следующие элементы:**

* Набор полей ввода для задания координат точки и радиуса области в соответствии с вариантом задания: Spinner {'-5','-4','-3','-2','-1','0','1','2','3'} для координаты по оси X, Text (-5 ... 3) для координаты по оси Y, и Spinner {'-5','-4','-3','-2','-1','0','1','2','3'} для задания радиуса области. Если поле ввода допускает ввод заведомо некорректных данных (таких, например, как буквы в координатах точки или отрицательный радиус), то приложение должно осуществлять их валидацию.
* Динамически обновляемую картинку, изображающую область на координатной плоскости в соответствии с номером варианта и точки, координаты которых были заданы пользователем. Клик по картинке должен инициировать сценарий, осуществляющий определение координат новой точки и отправку их на сервер для проверки её попадания в область. Цвет точек должен зависить от факта попадания / непопадания в область. Смена радиуса также должна инициировать перерисовку картинки.
* Таблицу со списком результатов предыдущих проверок.
* Кнопку, по которой аутентифицированный пользователь может закрыть свою сессию и вернуться на стартовую страницу приложения.

**Дополнительные требования к приложению:**

* Все результаты проверки должны сохраняться в базе данных под управлением СУБД Oracle.
* Для доступа к БД необходимо использовать Spring Data.



# Программа

[https://github.com/DiedCloud/ITMO/tree/main/Web/Lab](https://github.com/DiedCloud/ITMO/tree/main/Programming/Lab5/Project)4

# Заключение

В ходе выполнения лабораторной работы успешно реализованы: график, рисующийся динамически, в зависимости от выбранного радиуса; возможность добавления точки на него с помощью клика мыши; обработка вводимых пользователем значений как на стороне front end, реализованном на angular, так и в логике сервера (Java Spring Framework); хранение предыдущих запросов клиента, а именно добавленных точек, в базе данных под управлением PostgreSQL; аутентификация пользователей и работа с их данными.

# Ссылки

1. *border-radius*. (21 2 2023 г.). Получено из mdn web docs: <https://developer.mozilla.org/en-US/docs/Web/CSS/border-radius>
2. Frameworks, J. (02 10 2022 г.). *HTML5 Canvas basics tutorial using Angular 14 project*. Получено из youtube: <https://www.youtube.com/watch?app=desktop&v=nJJ-P6PYYGI>
3. *How to Fix "localStorage is not defined" in Next.js*. (б.д.). Получено из developer.school: <https://developer.school/snippets/react/localstorage-is-not-defined-nextjs>
4. *position*. (24 10 2023 г.). Получено из mdn: <https://developer.mozilla.org/en-US/docs/Web/CSS/position>
5. *Reactive Forms*. (15 11 2023 г.). Получено из METANIT: <https://metanit.com/web/angular2/5.5.php>
6. TV, U. (12 12 2020 г.). *Spring framework ПОЛНЫЙ КУРС для начинающих 2021 REST API*. Получено из youtube: <https://www.youtube.com/watch?v=q87Xxu4NPIc>
7. *ViewChildren*. (б.д.). Получено из angular: <https://angular.io/api/core/ViewChildren>
8. *Валидация форм¶*. (б.д.). Получено из Справочник Angular¶: <https://angdev.ru/archive/angular9/forms-validation/>
9. *ДОСТУП К КОМПОНЕНТАМ, ДИРЕКТИВАМ И ЭЛЕМЕНТАМ DOM В ANGULAR*. (02 03 2021 г.). Получено из 8host: <https://www.8host.com/blog/dostup-k-komponentam-direktivam-i-elementam-dom-v-angular/>
10. *Жизненный цикл компонента*. (13 11 2023 г.). Получено из METANIT: <https://metanit.com/web/angular2/2.8.php>
11. Минин, В. (19 07 2022 г.). *Angular 14 - Быстрый Курс [2022]*. Получено из youtube: <https://www.youtube.com/watch?v=yCIti018Srw>
12. *Создание атрибутивных директив*. (13 11 2023 г.). Получено из METANIT: <https://metanit.com/web/angular2/3.2.php>