

Университет ИТМО

Веб-программирование
Лабораторная работа №4

Работу выполнил:
Бавыкин Роман
Группа: Р3210
Вариант 10683

Санкт-Петербург
2022 г.

1. Текст задания.

Переписать приложение из [предыдущей лабораторной работы](#) с использованием следующих технологий:

- Уровень back-end должен быть основан на Spring.
- Уровень front-end должен быть построен на [React](#) + [Redux](#) (необходимо использовать ES6 и JSX) с использованием набора компонентов [React Toolbox](#).
- 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 1078 пикселей.
- "Планшетный" - для устройств, ширина экрана которых равна или превышает 805, но меньше 1078 пикселей.
- "Мобильный"- для устройств, ширина экрана которых меньше 805 пикселей.

Стартовая страница должна содержать следующие элементы:

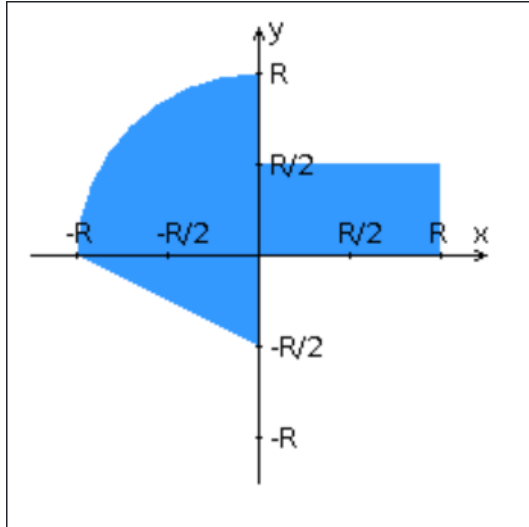
- "Шапку", содержащую ФИО студента, номер группы и номер варианта.
- Форму для ввода логина и пароля. Информация о зарегистрированных в системе пользователях должна храниться в отдельной таблице БД (пароль должен храниться в виде хэш-суммы). Доступ неавторизованных пользователей к основной странице приложения 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: Dropdown {'-2','-1.5','-1','-0.5','0','0.5','1','1.5','2'} для координаты по оси X, Input (-5 ... 5) для координаты по оси Y, и Dropdown {'-2','-1.5','-1','-0.5','0','0.5','1','1.5','2'} 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 зависеть от факта попадания / непадения 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 PostgreSQL.
- Для доступа к БД необходимо использовать Spring Data.



2. Исходный код программы:

<https://github.com/robqqq/Web4>

3. Разработанное Web-приложение:

Roman Bavykin, P3210, V-10683

Username *

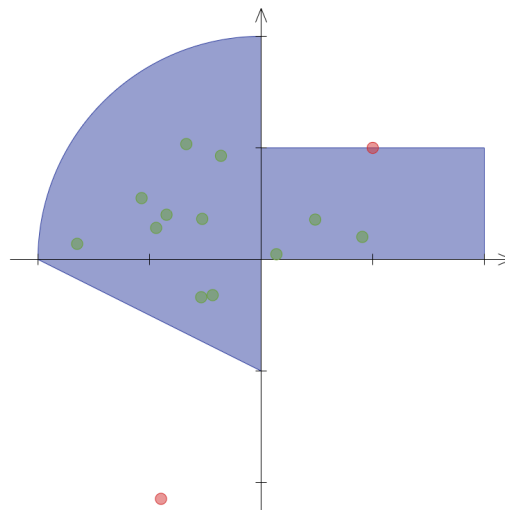
0/20

Password *

0/20

LOGIN

REGISTER



X
-0.5

Y
-0.5

R
2

2

+ ADD CLEAR

X	Y	R	Result
-0.5279503105590062	0.36335403726708076	1.5	True
0.13664596273291926	0.046583850931677016	1.5	True
-0.43478260869565216	-0.3198757763975155	1.5	True
1	1	1.5	False
1	1	1.5	False
1	1	1.5	False
1	1	1.5	False
-0.36024844720496896	0.9285714285714286	1.5	True
1	1	1.5	False
0.484472049689441	0.35714285714285715	1.5	True
1	1	1.5	False
0.906832298136646	0.20186335403726707	1.5	True

4. Выводы по работе.

Написал restful веб-приложение, back-end которого был реализован на Spring Boot с использованием модулей Spring Restful, Spring Web MVC, Spring Data, Spring Security, front – на react с использованием redux и набора компонентов react-toolbox. Изучил и укрепил такие понятия, как DI, IoC, также изучил работу js-фреймворков, научился настраивать webpack и с помощью него собирать react-приложение.