

## Задание “Телефонная книга”

### Требования

Язык программирования: Java

Инструменты:

JDK (<http://java.sun.com>)

Spring (<http://spring.io/>)

Maven (<https://maven.apache.org/>)

Хранимые данные

Информация о пользователе в системе

Логин (только английские символы, не меньше 3х, без спецсимволов)

Пароль (минимум 5 символов)

ФИО (минимум 5 символов)

Хранимая информация (одна запись)

Фамилия (обязательный, минимум 4 символа)

Имя (обязательный, минимум 4 символа)

Отчество (обязательный, минимум 4 символа)

Телефон мобильный (обязательный)

Телефон домашний (не обязательный)

Адрес (не обязательный)

e-mail (не обязательный, общепринятая валидация)

Задание:

Реализовать Web проект “Телефонная книга”. Содержащий страницы:

авторизацию

Вход в систему (логин/пароль)

Регистрация

Выход из системы

Работа с хранимыми данными

Просмотр всех данных с возможностью фильтрации по имени/фамилии и номеру телефона (не полное соответствие).

Добавление/Редактирование/Удаление хранимых записей

Система доступна только авторизованным пользователям. Если пользователь не авторизован, при попытке открытия любой страницы его должно редиректить на страницу авторизации. На странице авторизации он может ввести логин и пароль для входа в систему или зарегистрироваться. При регистрации указываются поля: ФИО, логин и пароль.

У каждого авторизованного пользователя имеется своя телефонная книга, т.е. каждый пользователь видит только те записи, которые он создал.

Обратить внимание (обязательно к выполнению)

Админка для управления пользователями - не требуется.

Формат телефонов должен проверяться и быть валидным для Украины, пример: +380(66)1234567

Приложение обязательно должно содержать JUnit тесты, максимально плотно покрывающие код.

Приветствуется использование Mockito.

Проект должен собираться средствами Maven

Для запуска использоваться SpringBoot

Исходный код должен быть выложен на GitHub/BitBucket

Все настройки приложения должны находится в properties файле, путь к которому должен передаваться в качестве аргументов JVM машине (-Dlardi.conf=/path/to/[file.properties](#)).

В конфигурационном файле указывается тип хранилища. Тип хранилища используется один раз при старте JVM (изменения в конфигурационном файле вступают в силу только при перезапуске JVM). Реализовать минимум два варианта хранилища: СУБД (MySQL) и файл-хранилище (XML/JSON/CSV на выбор). Настройки хранилища должны указываться в файле-конфигурации (хост и пользователь для СУБД или путь к файлу для файлового хранилища).

Для файлового хранилища - в случае отсутствия файла(ов) - его(их) необходимо создать. Для СУБД-хранилища в файле README.md должен находится SQL запрос для создания всех необходимых таблиц.

Проверка данных должна осуществляться на стороне сервера.

Приложение должно содержать четкое логическое разделение между представлением, логикой и источником данных.

Приветствуется

использование JQuery, Bootstrap