

## Тестовое задание на позицию Junior Java Developer

Разработать систему управления пользователями.

Система должна представлять собой WEB-приложение, которое предоставляет интерфейс управления пользователями.

Система должна поддерживать следующую ролевую модель

Роль	Доступные действия
USER	Авторизация (Login) Просматривать список пользователей (List) Просматривать детали любого пользователя (View) Выход (Logout)
ADMIN	Авторизация (Login) Просматривать список пользователей (List) Просматривать детали любого пользователя (View) Создание нового пользователя (New/Edit) Редактирование существующего пользователя (New/Edit) Lock/Unlock пользователя (Lock/Unlock) Выход (Logout)

Основная сущность в системе – UserAccount.

Field	Description	Length	Validation
Username	Имя пользователя для авторизации.	3-16	Только латинские символы; Должна быть уникальным
Password	Пароль пользователя	3-16	Только латинские символы и цифры. Минимум один символ; Минимум одна цифра;
First Name	Имя.	1-16	Только латинские символы
Last Name	Фамилия	1-16	Только латинские символы
Role	Роль пользователя USER or ADMIN На UI должно быть dropdown.		
Status	Состояние пользователя ACTIVE or INACTIVE На UI должно быть dropdown.		
Created At	Дата заведения пользователя. Проставляется автоматически при записи в БД.		

Пароли в БД хранить в хешированном виде с применением алгоритма bcrypt.

В системе должны быть следующие страницы:

Страница	URL	Описание	Доступные действия (с учетом роли)
Login	/login	Авторизация в системе	Вход в систему; Если состояние пользователя INACTIVE, то вход невозможен;

			Если логин/пароль не верен, то вход невозможен;
List	/user	Список пользователей	Опционально: <ol style="list-style-type: none"> <li>1. Фильтрация списка по Username</li> <li>2. Фильтрация списка по Role</li> <li>3. Пагинация</li> </ol>
View	/user/{id}	Просмотр деталей пользователя	Lock/Unlock: должно приводить к смене состояния ACTIVE/INACTIVE
New	/user/new	Создание нового пользователя	При наличии валидационных ошибок должно показываться соответствующее сообщение на UI.
Edit	/user/{id}/edit	Редактирование пользователя	При наличии валидационных ошибок должно показываться соответствующее сообщение на UI.

Предпочтительный стек технологий:

1. Java 8+
2. Spring (Spring Boot, Spring MVC, Spring Data, Spring Security)
3. PostgreSQL
4. Html Template Engine - любой (Thymeleaf, Freemarker, etc.)
5. Gradle
6. Опционально: сборка docker-образов

Можно использовать сторонние библиотеки