Summary

Create a Spring Boot RESTful Web Service application that will be used to manage two data types:

- **User:** A user of the system
- **Phone:** A phone belonging to a user. Each user can have 0 to many phones.

Both types will need to be persisted by the application in the storage medium of your choice. A good starting point for this exercise is spring initializr. This will generate a starter Spring Boot project for you. The output of this exercise can be checked into a repository or bundled up and sent to us directly. The exercise will be discussed in the interview.

A full and comprehensive implementation is not expected to be delivered. Focus on what you believe is most important.

Data Types

The following are representations of the two data types. These are not final and can be tailored to suit your application as needed.

User:

```
{
   "userId":"93f3ed0a-92bd-4c82-ba0e-c098111cef59",
   "userName":"john_user",
   "password":"*****",
   "emailAddress":"john@example.com",
   "preferredPhoneNumber":"+353881234567"
}
```

- `phoneId` is a `UUID`
- `userName` is a `String`
- `password` is a `String`
- `emailAddress` is a `String`
- `preferredPhoneNumber` is a `String`

Phone:

```
{
   "phoneId":"2a810635-18d0-40ce-ada6-0c0fd1181225",
   "phoneName":"John's Pixel",
   "phoneModel":"ANDROID",
   "phoneNumber":"+353881234567"
}
```

- `phoneId` is a `UUID`
- `phoneName` is a `String`
- `phoneNumber` is a `String`
- `phoneModel` can be any of: `[IPHONE, ANDROID, DESK_PHONE, SOFT_PHONE]`

Application Functionality

The following are tasks that the application could support:

- Add a user to the system
- Delete a user from the system
- List users in the system
- Add a phone to a user
- Delete a user's phone
- List a user's phones
- Update a user's preferred phone number

Things to consider for your application

- Testing
- Logging
- Basic Authentication & Authorization if appropriate
- Documentation
- Running instructions