**Specification Document:**

**Product Capabilities:**

1. Admin is provided with a login to start the service.
2. Admin has the ability to change the login password.
3. Once logged in, admin can Create, Retrieve, Delete and modify records of a shoe product, subscribed users (in the prototype) and purchase(also in the prototype).
4. In real scenario admin has the ability to categorize shoe products, see logged in users and check the purchase reports.

**Appearance**:

1. Login page is generic, console/ command line interface to avoid creating a front end in the current phase.
2. The admin can visit “localhost:2020/h2” to see the data values as database form with the ability to use SQL queries.
3. Users can use any API testing tool such as postman to fetch, insert, delete or update data.

**User interactions:**

1. At the beginning, admin is prompted to enter default password: “admin”.
2. Then he/she has the option to change the password or launch the spring boot project.
3. Once launched, “localhost:2020/h2” gives users the option to use SQL queries on “Shoes”, “subscriber” and “ProjectReport”.
4. Also as per the admin’s choice can use an api testing tool for the same.
5. localhost:2020, \<value> gives the user various options:
6. \<object name>: list all objects
7. \<object name>\<column-name>/<value>: retrieves objects with particular column value.
8. Here key objects are Shoes, Subscribers, Project Report

Stopping the spring application ends the application.

**Git**

Here is the github link:

<https://github.com/Bodha/SportyShoes/tree/master>

**Java Concept:**

1. @SpringBootApplication : to enable a host of features, e.g. Java-based Spring configuration, component scanning, and in particular for enabling Spring Boot's auto-configuration feature.
2. @RestController: used in the classes as controllers where @RequestMapping methods are used for @ResponseBody semantic.
3. @RequestMapping for get, post, put etc. http/https requests.
4. @ResponseBody refers to the value or semantic to pass.
5. @Autowired: Marks a constructor,methods, or config method as to be autowired by Spring's dependency injection facilities.
6. Lombok to automatically assign getter, setter, Tostring, etc functionality to a base class.
7. H2: a persistent database for the duration of an active spring app. Also called embedded database, provides SQL queries to be written for it.
8. @Repository : A bit similar to implementing DAO. Provides mechanisms for encapsulating storage, retrieval, and search behavior that has scope over a range of objects.
9. @Service: Associated with the service layer of a project, with no encapsulated state.
10. Application.properties in resources help to set spring properties.
11. Pom.xml file in maven helps update dependencies, properties etc on the go.