**Online E-Health Care Management System**

**Existing System**

The customer goes to the shop and purchases the medicine required. So a lot of time is wasted and the person gets tired. If he wants to exchange the product, once again he goes to the shop and replaces them. The complete process depends on the physical interactions.

**Proposed System**

As a result, the team decided to hire a Full Stack developer to develop a healthcare web application with a rich and user-friendly interface.  
You are hired as the Full Stack Java developer and are asked to develop the web application. The management team has provided you with the requirements and their business model so that you can easily arrange different components of the application.

The customer selects the required medicines and orders them by a single click. Before it, the customer needs to create a login account and fill all the details like name, address, any id no…Etc.  The client can able to view the status of the medicines. The business goal for the application is to provide the medicines to all the people & admin will provide the supplier details.

**Project Statement.**

Create a dynamic and responsive Java e-healthcare web application for ordering medicines of different categories.

**Recommended technologies:**

1. Database management: MySQL
2. Backend logic: Spring boot with Spring Data

**Project development guidelines:**

* The project will be delivered within three sprints with every sprint delivering a minimal viable product.
* It is mandatory to perform proper sprint planning with user stories to develop all the components of the project.

**1st Sprint**

Using Spring boot with Spring Data/ JPA developer has to developer has to create REST API.

There will be two portals in the application, namely admin and user portal.

Sample URL for the REST API for backend technologies is

<http://localhost:8080/api>

<http://localhost:8080/api/admin>

**Admin**

* 1. **Sign In** :
     + Admin will sign in with the already pre-entered credentials present in the database
  2. **Add medicine** 
     + Admin can add the new medicine with details like medicine name, company name, price, quantity, uses and expire date with the auto-generated medicine id and etc if necessary
  3. **Update medicine** 
     + Admin can update the price, quantities and company name
     + Discounts for the particular medicine if required
  4. **Delete medicine** 
     + Admin can delete the medicine which has no demands (zero demand in the market)
     + If medicine expired.
  5. **View customer/user details** 
     + - Admin can view customer details like name, contact number, address.
  6. **Generate Reports**
     + Admin can view the reports that are ordered by users
     + Admin must able to customize the reports like
       - Stock reports
       - Sales reports
       - Medicine reports
       - Range report like
         * Monthly reports
         * Weekly reports
         * yearly reports
  7. **Logout**
     + Log out from the application

<http://localhost:8080/api/user>

User

**1. Sign up:**

* + - Admin can view the reports that are ordered by user
    - User can signup entering details like firstname, lastname, email, password, dob, phone no, address and user id should be auto-generated
    - Once user registers, user should able to see the auto-generated id using that user can sign in
    - Once user registers there should be some default amount to purchase items like 1000.
  1. **Sign in:**
     + Sign in will take user id and password, on successful authentication go to the dashboard of the user else show the error messages and allow user to re-login
  2. **Search medicine:**

User must able to search medicine by uses or disease

* 1. **Select medicine:**
     + User must able to add medicine to the cart
  2. **Delete items:**
     + User must able to delete medicine from the cart
  3. **Update items:**
     + User can update the quantities
  4. **View items in the cart:**
     + User could able to see all the items added in the cart
  5. **Checkout:**
     + Once all the items are selected user can checkout to order the items, which will deduct the amount if the money is present in the funds else user gets the error message insufficient amount, if the sufficient money is present in the funds, order must be placed successfully
  6. **Order Status:**
     + User can see the history of orders and their status like delivered, shipped, out for delivery and etc
  7. **Edit Profile:**
     + User must able to edit the profile like password, address, phone number, email id and etc
  8. **Funds:**
     + Funds will show the available amount to purchase the items
     + User can add money by entering the account number and amount, all the users will have some account number with balance in their dummy bank account.
     + The amount user adds will be added to the profile funds which is utilized at the time of ordering the items
  9. **Logout:**
     + Logout from the application

**Points**

1. So in Sprint1, you have to create REST API based upon the above requirement.
2. So you have to use proper http protocol methods like GET, POST, PUT and DELETE depending upon the type of operation in admin as well as customer/user details.