**LOW LEVEL DESIGN**

**Problem Statement:**

Measurements differ for different Brands for a particular size, so when a user purchases an apparel from a particular brand of a size, he should be recommended with the appropriate size when he switches to another brand next time.

**Project API:**

1. **Registration for new Users:**

This API takes Name, Mobile Number, Password and Gender as Input and saves to the user Table. User can be deleted using userId.

1. **Login With Validation Check:**

This API takes Mobile Number as username and password as input for user Authentication. Then it's base64 encoded in String format and set as value of Authorization, finally it's passed as Request header.

1. **Show Brand List with Size:**

This API takes Purchase Category and its Dimensions as input and shows the respective size of all the available brands. If the user does not wish to enter the dimension then the user history is used to get the dimension and the above task is completed.

1. **Show Interested Brand Size:**

This API takes Category, interested brand, reference brand and size as input and shows the size of the interested brand. If the user does not wish to enter the reference brand or size then the user’s previous purchase is used for reference and the above task is completed.

**Tables and their description:**

1. **User table:** It contains the user details like UserId, Name, Mobile Number, Password and their gender. The table gets updated each time a new user register. The structure of the table is as shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Column names of table** | **Type** | **Mandatory field** |
| 1 | Id(primary key) | int | Yes |
| 2 | Name | String | Yes |
| 3 | Mobile Number | String | Yes |
| 4 | Password(Hashed) | String | Yes |
| 5 | Gender | String | Yes |

1. **Brand Table:** It contains the brand details like BrandId, BrandName and enable. BrandId acts as foreign Key in Other Tables. Enable is used to determine if a Brand is available or not.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Column Name** | **Type** | **Mandatory** |
| 1 | Brand Id (Primary Key) | Int | Yes |
| 2 | Brand Name | String | Yes |
| 3 | Enable | Int | Yes |

1. **Category Table:** This table contains the list of Categories and their Unique ID (CategoryId). CategoryId is used in other tables as foreign key and it maps to a particular Category.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Column Name** | **Type** | **Mandatory** |
| 1 | CategoryId  (Primary Key) | Int | Yes |
| 2 | Category Name | String | Yes |

1. **Size Table:** This table contains sizeid, CategoryId and dimensions of all the brands.

This table cannot be updated by the user. It is updated when a new brand is added.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Column name** | **Type** | **Mandatory** |
| 1 | SizeId(Primary Key) | Int | yes |
| 2 | CategoryId(Foreign Key) | Int | Yes |
| 3 | BrandId (Foreign key) | Int | Yes |
| 4 | MaleHeight | Int | No |
| 5 | MaleChest | Int | No |
| 6 | MaleSleeves | Int | No |
| 7 | MaleWaist | Int | No |
| 8 | MaleShoeSize | Int | No |
| 9 | FemailHeight | Int | No |
| 10 | FemaleChest | Int | No |
| 11 | FemaleSleeves | Int | No |
| 12 | FemaleWaist | Int | No |
| 13 | FemaleShoeSize | Int | No |
| 14 | Size | String | Yes |

1. **Order History:** This table contains the user purchase details. It has 2 Foreign keys

Userid and customerid. This table gets updated when the user makes a new purchase.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Column Name** | **Type** | **Mandatory** |
| 1 | Id (Primary Key) | Int | Yes |
| 2 | userid(Foreign Key) | Int | Yes |
| 3 | Categoryid(Foreign Key) | Int | No |
| 4 | Sizeid (Foreign Key) | Int | No |

**Flow Diagram:**

