**TECH CHALLANGE**

**JAVA**

**Description**

Build an API to:

* Allow Authentication;
* Query Products;
* Receive Orders;
* Cancel an Order;
* Get Order Status;
* Store data in a database of his/her choice;

**Technologies**

**Java** will be used as the back-end language.

**Maven** framework will be used to manage dependencies.

**Junit** framework will be used to handle unit tests.

**Spring** framework will be used to create the WebServices.

**log4j** framework will be used to handle logs.

**MySQL** will be used as a database.

**Java.sql** will be used to manage connection and transactions between the Java and MySQL.

**Database**

Database name = “techChallengeDb”;

Database tables

* + Customer
  + Order
  + OrderItem
  + Product
  + Restaurant
  + OrderReview (TBD)

**Backlog**

* Create a database using mysql
  + Create an customer table to store users to authenticate
  + Create an order table to store orders
  + Create a status table to map all possible order status
  + Create a restaurant table
  + Create a products table
  + Create a order review table (TBD)
* Create a java project
  + Create model objects to reflect database tables (Data transfer objects)
    - AuthenticationDTO
    - CustomerDTO
    - OrderDTO
    - OrderItemDTO
    - RestaurantDTO
    - ProductDTO
    - OrderReviewDTO (TBD)
  + Create Util package
    - OrderStatusEnum
    - Util
  + Create test layer
    - Create unit test to validate webservices
      * AutheticationDTO UserAuthentication(String email, String password)
      * List<RestaurantDTO> getRestaurants(float userLatitude, float userLongitude)
      * List<ProductDTO> getProductsFromRestaurant(int restaurantId)
      * OrderDTO createOrder(CustomerDTO user, List<ProductDTO> productList)
      * OrderStatusDTO cancelOrder(Integer orderId)
      * OrderStatusDTO getOrderStatus(Integer orderId)
  + Create Business layer
    - to authenticate users
    - to get restaurants from database
    - to get products from database
    - to create order
    - to cancel order
    - to get order status
  + Create web services
    - AuthenticationDTO UserAuthentication(String user name, String password)
    - List<RestaurantDTO> getRestaurants(float userLatitude, float userLongitude)
    - List<ProductDTO> getProductsFromRestaurant(int restaurantId)
    - OrderDTO createOrder(CustomerDTO user, List<ProductDTO> productList)
    - OrderStatusDTO cancelOrder(Integer orderId)
    - OrderStatusDTO getOrderStatus(Integer orderId)