## Advanced technology Final practice

Seoyoung ko alu.109223

20/1/2020

1. **Introduction**

"Amazon" like e-commerce:

There will be administrators (they don't sign up, they already exist in the system).

Many stores will sell through this e-commerce as if they were a single store.

**Administrators** can:

\* Manage the categories the whole e-commerce. (Only 1 level of categorization) (Eg Electronics, Videogames, Home, Clothing, Sports, etc ...)

\* Consult all sales of a given day

**Stores** can:

* Sign up (store name, email, password)
* Manage the products they sell:
* Create products (name, description, category, price)
* Add / remove stock of their products
* Check their sales for a given day

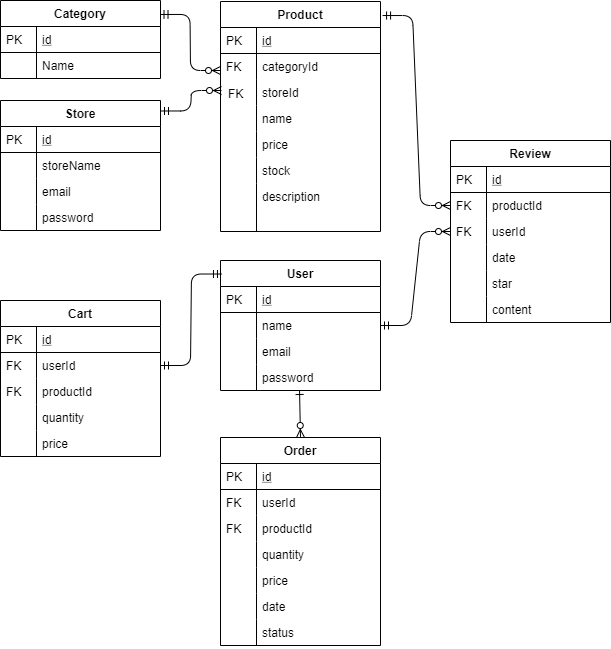
**Users** can:

* Sign up (name, email and password)
* Query products:

1. By category
2. By price range
3. The store is not relevant to the search, it doesn't matter the store that actually sells the products. (Nobody will search products by store)

* Add reviews to a product
* Query product reviews
* Add products to the cart (there is only one cart per user and it is saved until confirmed or canceled)
* Confirm the purchase.
* The purchase of a product will trigger the sending of an email (dummy send, you can simulate it by writing on the console). But that dummy send will be programmed in a class called EmailService in the email.service.ts file and that you should use during the purchase confirmation.

1. **Object model design**



* Category
  + id : string
  + name : string
* Store
  + id : string
  + storeName : string
  + email : string
  + password : string
* User
  + id : string
  + name : string
  + email : string
  + password : string
* product
  + id : string
  + categoryId : string
  + storeId: string
  + name : string
  + price : number
  + stock : number
  + descriptiton : string
* cart
  + id : string
  + userId : string
  + productId : string
  + quantity : number
  + price: number
* order
  + id : string
  + userId : string
  + productId : string
  + quantity : number
  + price: number
  + date : string(timestamp)
  + status : number
* review
  + id : string
  + productId : string
  + userId : string
  + date : string(timestamp)
  + star : number
  + content : string

1. **Model URIs design**

|  |  |  |
| --- | --- | --- |
| URI | Description | Auth |
| < Category > | | |
| POST /categorys | Create a new category | Admin |
| PUT /categorys/:id | Update name of category | Admin |
| GET /categorys | Retrieve all categorys |  |
| GET /categorys/:id | Retrieve a category by id |  |
| DELETE /categorys/:id | Delete a category by id | Admin |
| < Sale > | | |
| GET /sales | Retrieve total sales | Admin |
| GET /store/:storeId/sales | Retrieve sales of a store by storeId | Store |
| < Product > | | |
| POST /stores/:storeId/products | Store create a new products | Store |
| PUT /stores/:storeId/products/:id | Update a existing product information | Store |
| GET /products/:id | Retrieve a product by id |  |
| GET /products | Retrieve all product |  |
| GET /products?category={cid} | Filtering products by given category id |  |
| GET /products?min=0&max=30 | Filtering products by price |  |
| PATCH /stores/:storeId/products/:id | Store add or delete stock of product | Store |
| < Cart > | | |
| POST /carts | Create a new cart | User |
| GET /carts | Retrieve all carts |  |
| GET /carts/:id | Retrieve a category by id |  |
| GET /user/:id/carts | Retrieve a category by user id |  |
| DELETE /carts/:id | Delete a category by id | User |
| < Order > | | |
| POST /orders | Create a new order | User |
| PUT / orders | Update a order status | Store |
| GET /orders | Retrieve all orders | Store |
| GET / orders /:id | Retrieve a order by id |  |
| GET /user/:id/ orders | Retrieve a order by user id | User |
| < Review > | | |
| POST /reviews | Create a new review | User |
| PUT /reviews/:id | Update a description of review | User |
| GET /reviews | Retrieve all reviews |  |
| GET /reviews/:id | Retrieve a review by id |  |
| GET /reviews?productId=pid | Find reviews by product id |  |

|  |  |
| --- | --- |
| < User > | |
| POST /users | Create a user |
| PUT /users/:id | Update name of user by id |
| GET /users | Retrieve all users |
| GET /users/:id | Retrieve a user by id |
| GET /users/:email | Retrieve a user by email |
| DELETE /users/:Id | Delete a user by id |
| < Store > | |
| POST /stores | Create a store |
| PUT / stores /:id | Update name of store by id |
| GET / stores | Retrieve all stores |
| GET / stores /:id | Retrieve a store by id |
| GET / stores /:email | Retrieve a store by email |
| DELETE / stores /:Id | Delete a store by id |