Cloud Project Report

Ιωαννίδης Χρήστος (2018030006)

Page IP: 34.78.204.160
Mail: cioannidis@tuc.gr

All the requested functionality has been implemented:

- **Keycloak authentication-authorization functionality (Postgres Database)**
- **Frontend**
- **Product Service (mySql Database)**
- **Order Service (mySql Database)**
- **Publish-Subscribe Mechanism (kafka)**
- **All run with docker**
- **Pages Can Only Be Accessed By Corresponding Users**
- **Deployment On Google Cloud**

Brief API Explanation:

- **Frontend**

GET	**/**	**Login page**
GET	**/products_client**	**Main page for customer**
GET	**/products_seller**	**Main page for seller**
GET	**/orders_page**	**Page where a customer can view his/her orders**
GET	**/cart**	**Customer cart page**
GET	**/edit_product**	**Page where seller can edit product details**
GET	**/new_product**	**Page where seller can create a new product**

- **Product Service**

PUT	**/edit_product/:productId**	**Renew product details (by id)**
GET	**/products_all**	**Return all products**
DELETE	**products/:productId**	**Delete product (by id)**
GET	**/products/:username**	**Get all the products by seller name**
GET	**/search/:searchTerm**	**Get all products that look like searchTerm**
POST	**/products**	**Create a new Product**

- **Order Service**

GET	**/orders**	**Get all orders (used for testing)**
GET	**/orders/:user**	**Get all orders of a specific customer**
POST	**/orders**	**Create order**

Some sample users

Customers:	Username: customer , Password: customer
	Username: c , Password: c
Sellers:	Username: s , Password: s
	Username: seller , Password: seller

Project Notes:

- Some menus are not exactly resounding examples of human-machine interaction (e.g. some actions like when the submit order button is pressed,it doesn't display anything to give the user the affirmation that all went ok)
- Some wait-to-start timers were set for the Database services and kafka connectors in order to attempt connection after all the other services they depend on have started and are functioning normally. This is a factor that should be taken in account depending on the speed of the machine they are being installed on. I doubt that this is the best way to implement such functionality, instead, there should have been some event signal for when the other services are ready for action, however i couldn't find such signals so i ended up with the implementation described above.