**Project-Report**

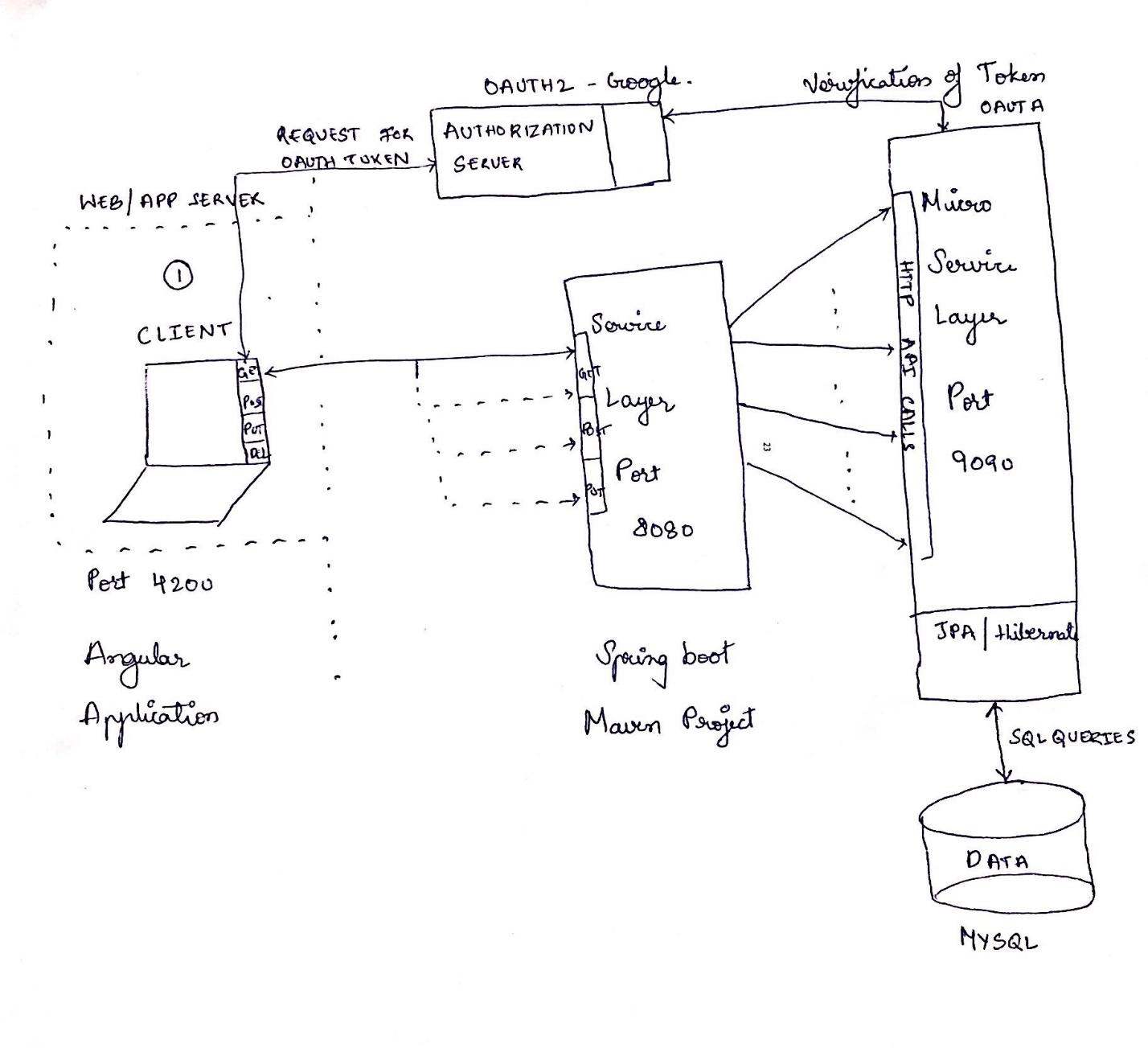
**Web Programming Languages (CS-6314)**

**Fall- 2017**

Group members: Jahnavi Mesa, Md. Ather Shareef, Shashank Kumar Som

**Architecture:**

R**esponsive web site** and **scalable web application** based on the **Service Oriented Architecture** (SOA). This project make use of **Web-Services** and **Micro-services** for implementing the SOA.

****

**Front-End (Angular 5)**

Our front-end client is developed using Angular 5 framework, which is a complete framework for web UI development. Front-end server runs on a separate server controller by Angular 5 framework.

Why Angular?

1. Angular is a faster and smaller framework to develop responsive website.
2. Angular introduces a new view engine system, which can reduce the size of generated code for our components by around 60%.
3. Angular helps to develop 1 page web applications that load components dynamically thus reducing the band-with consumption.
4. Built-in animation package to perform complicated animations which otherwise would require heavy coding.
5. Ability to include other responsive frameworks like bootstrap etc.

**Service Layer:**

Service layer is designed using SpringBoot framework along with maven dependency management tool and the service layer is hosted on a separate apache tomcat server.

**Micro-Service Layer:**

Similar to the service layer, Micro-service layer is also designed using SpringBoot framework and maven and is hosted on a separate apache tomcat server. It uses JPA/Hibernate framework to communicate with the database server.

**Database Layer:**

MySQL is used as the database server hosting the data generated by our web application. Again, JPA is used to communicate with the Database and the communication is SSL enabled and secure.

**Requirements fulfilled:**

A. **HTML/CSS/JavaScript**: Web site’s client side Graphical User Interface (GUI) is built using Angular 5 that incorporates HTML/CSS/Typescript.

B. **Server-side Programming**: Java programming language is used for the website’s server-side implementation and Web Services & Micro-services implementation.

C. **Web Application Domain and Functionalities**: The Web Application implements a gift registry creation and sharing web application similar to the ones available on Amazon.com, Target.com, etc.

The web application support the following functionalities via a web browser based Graphical User Interface (i.e. webpages):

1. New regular user registration

2. Regular User

i. Login

ii. Logout

iii. User profile information display and editing

iv. Forgot password functionality

v. Ability to create a registry

vi. Ability to search for items that you would like to add into registry

3. Table display:

a. Results (with at least four properties) should be displayed in a sortable table (i.e. allowing resulting to be sorted on any column)

4. Search results filtering capabilities on at least four result item properties

vii. Ability to add/remove items from a registry

viii. Ability to share registry to particular user or make it public

ix. Self-assign an item on another user’s registry

5. Accessible any unavailable page should retrieve a pretty and generic 404 page

6. Admin User:

i. Login

ii. Logout

iii. Add/remove items into/from inventory

iv. Display the items in the inventory

**Functionalities Supported:**

The web application supports the following functionalities via a web service API:

1. New regular user registration

2. User profile information access and modification

3. Ability to create a registry

4. Ability to search for items in the inventory

5. Ability to add/remove items from a registry

6. Ability to share registry to particular user or make it public

7. Self-assign an item on another user’s registry

8. Add/remove items into/from inventory

9. Display the items in the inventory

**Other Features Supported:**

Our web site/application implementation also includes the following four (4) features:

1. High Performance: Performed distributed caching. Ehcache which is an open-source alternative for implementing a distributed caching mechanism is used.

**Cache miss and cache hit information is available in the web/app server logs**.

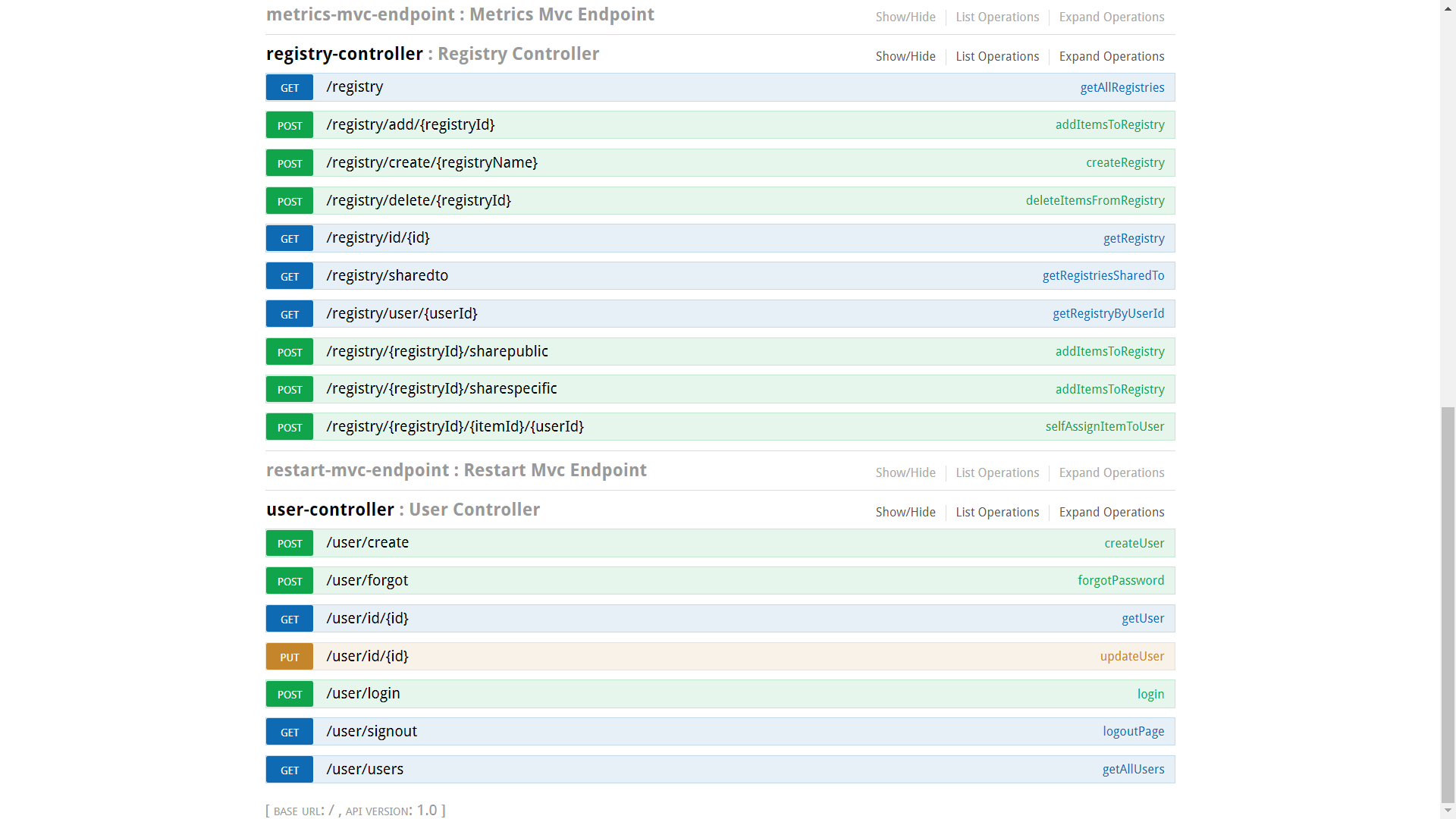
2. Client-Server Communication Encryption: communication channel is encrypted between the client (i.e. browser), web site server, Web Services, and Micro-services server using TLS/SSL.

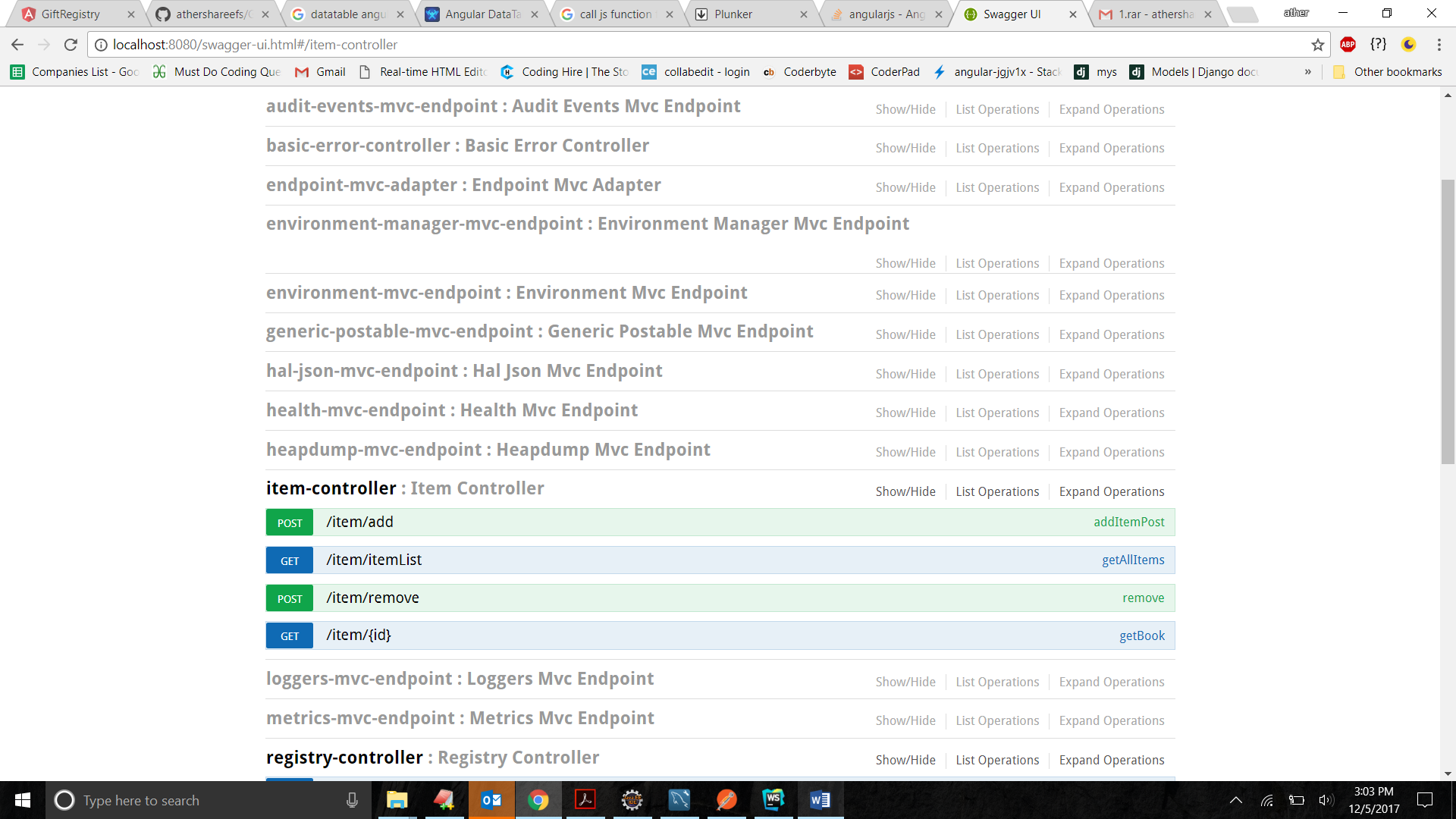
**This feature is implemented on the Website web/app server by using HTTPS protocol.**

3. Request/Response Compression: All calls between the modules in the architecture above are compressed and can be verified by examining the Header of the request and response.

**Extra Credit Features**:

1. Single Sign-On: Single sign-on oAuth is also implemented in our Web application.
2. **Services Layer API calls**





1. **Micro Services Layer API calls**

