

## **Middleware Architecture**

### **Assignment 2 – REST API**

**Note:** This assignment carries 20 marks. This is a group project. Maximum group size is 4. Optionally, you may have a lesser group size or you can do it individually (marking standard will be the same irrespective of the group size).

Assume that you have been asked to develop a collaborative shopping platform. Following are the requirements given by the client and/or the Business Analyst.

- The system should have a web interface where buyers can shop for items uploaded by sellers.
  - A service should be there where sellers can add/update/delete items.
  - A service should be there where buyers can search/buy items.
  - A buyer may buy multiple items.
  - Once an item or a collection of items are bought, the buyer may select the delivery option, where a request may be sent to a delivery service (so there should be a third-party delivery service).
  - The payment for the items bought can be made using credit cards or using the mobile phone number (which would be added to the user's mobile bill).
  - The system can connect to a payment gateway for credit card transactions. The information that should be submitted includes the credit card number, amount, CVC number (3 digit no. at the back of the credit card) and card holder's name.
  - The users can choose to make the payment using a mobile service provider's service to credit the mobile bill. The information that should be transferred includes the mobile phone number, a four digit pin number and the amount to be charged.
  - Once the payment is made the user should be given a confirmation of the appointment via SMS and email.
- 
- Based on the above information, develop a set of RESTful web services to implement the system (you may use any technology to implement the services).
  - Use the WSO2 EI (Enterprise Integration – ESB) to integrate services at the backend and expose a common web API.

For example, you can do some transformation at the EI to route the payment to either the banking payment gateway or the mobile operator, based on some parameter of the payment request message.

**Hint:** Refer the following documentation on ESB service integration for a guide to do this.

<https://docs.wso2.com/display/EI660/Routing+Requests+Based+on+Message+Content>

You can expose the rest of the services also through the WSO2 EI to the client. The advantage of this would be that the client(s) will see the same web API and do not have to access different services at the back-end. The WSO2 EI will route each request to the relevant service at the back-end.

- Develop an Asynchronous web client, using which the users may access the system. You may use any Javascript framework that supports asynchronous programming (Angular, React, etc.) to do this. You can also use regular JQuery + AJAX to develop the client.

Since there's a REST API in the backend, other types of clients (e.g. mobile clients) can reuse the backend business logic easily in the future. However, for the scope of the assignment, implementing just an asynchronous web client is sufficient.

- Use appropriate security/authentication mechanisms to uniquely identify each user and to authenticate each user. There should be two roles, buyer and seller.

#### **Deliverables**

- Source code of the RESTful Web Services and the web client in a git repository.
- Source code of the web client.
- WSO2 Enterprise Integration project (developed using Eclipse Developer Studio).
- A readme.txt document, listing down the steps to deploy the above deliverables.
- Members.txt file, containing the names, index numbers and the email addresses of the group members.
- Any database scripts or any other data-store documents (xml documents, flat files, etc.) that you may have used to store the sample data (e.g. shopping item details).
- An 10-12 page report in pdf format. The report should include a high level architectural diagram showing the services and their interconnectivity. Also, it should list out the interfaces (NOT the user interfaces, but the service interfaces) exposed by each service and should briefly explain each of the workflows used in the system (you may use diagrams to do this). You can also include the details about the authentication/security mechanisms adopted.
- 10 min (maximum) youtube video of the demo.

You may use code snippets in the report to explain the above.

The report must have an appendix with all the code that you have written (**excluding the auto-generated code**). **Do not paste screenshots of the code in the appendix and copy the code as text. If screenshots are added in the appendix, marks will be reduced.**

The report should be no less than **10 pages (excluding the appendix)** of length. **The report is the main component that should be marked. However, the code should be there to validate the implementation.**

**Note:** You may **implement dummy services** to simulate the payment gateway, mobile payment gateway and the delivery service. For email and SMS notifications, you may try to use an available service on the Internet. If you cannot find one, you can use dummy services to implement those as well. **For instance, the dummy payment gateway service can accept the**

**relevant set of input parameters and just return a message saying the “payment successful”, rather than doing an actual payment. You can implement similar dummy services for the SMS, email, mobile payment and delivery services.**

**Note:** All reports will be uploaded to Turnitin for plagiarism checking. If the turnitin similarity is above **30%**, **10%** of the marks will be reduced. For **50%** similarity, **50%** of the marks will be reduced. For reports with **80%** similarity, **no marks** will be given.

**Submission:** All files should be uploaded in a single zip archive. Only one member needs to upload the submission. All members will receive the same group mark.

Use the following directory structure the upload the answer. The compressed file name should be assign2\_submission.zip.

- Readme.txt with the link to the github repository (at the top of the file) containing the source code of the Services (contains the REST services), Client (Web client source files), WSO2 Enterprise Integration Eclipse project (developed using developer studio), database scripts, xml files or any other source/configuration/resource files. Make sure to maintain a proper directory structure in the github repo to easily locate each source file category.
- The Readme.txt file should also contain the link to the youtube demo video.
- The Readme.txt should also contain any instructions required on how to deploy the deliverables).
- Members.txt
- Report (including the appendix) in pdf format.

**Marking rubric is given below.**

#### **Marking Rubric**

Following rubric will be used in evaluating the assignment.

<b>Criteria</b>	<b>Good (10-8)</b>	<b>Average (4-7)</b>	<b>Poor (0-3)</b>
Application of SOA principles in the architecture and the design			
Having clearly defined service interfaces, that facilitate reusability			
Quality and the readability of the code, with meaningful and detailed comments.			

Integration of services using the Enterprise Service Bus (ESB)			
Comprehensiveness and the quality of the report			