COMP 4350 Project Iteration 2

Group E (PartEZ)

Implementation and deployment: **(4.5/7)**

- End-to-end, fully working (back-end, front-end, DB, and AWS) features for the user stories planned for iteration 2. **(1.5/4)**

You created service but you didn't consume it from the web application. You are just directly using the back-end. You must consume that service from your web site. Other it won't be a SOA. (-2)
All user stories should be in perspective of users. Instead of writing "Message board available for each event for users to communicate with other invitees and the", you write "As an invitee, I want to be able to chat with other invitees of the same event".
User1 created a public event. When User2 login, it doesn't show that event in public event list.
"Any logged in user can rsvp to a public even"- does not work. (-0.5)

- Correctly deployed service on AWS **(3)**

Testing: **(8/10)**

- Back-end unit tests for the new and old classes **(1/2)**

You should test business logics that are implemented in different classes. You are supposed to call business logic methods from test methods. In the previous iteration you had no separate business logic layer and you tested your model classes. In the current iteration you have not separated business logics from controllers and writing test for model class are not sufficient since methods in the model classes are framework provided.

Some tests are buggy and there is a fatal error too.

```
E...E....................
Fatal error: Call to undefined method
Illuminate\Foundation\Application::instsance() in C:\wamp\www\partEZ-
master\partEZ\tests\databasemocks\EventControllerTest.php on line 26 ...
```

- Front-end unit tests (e.g., using Jasmin) **(3/3)**

- At least one integration test per detailed story **(1/2)**

Only model classes are tested. No business logic class found. (-1)

- Selenium. at least one automated end-to-end test per user story. **(3/3)**

You must automate Selenium test.

Design and documentation: **(4/5)**

- Good programming practices **(1/1)**

Bad namespace for model classes. It should be “App/Models/Event” instead of “App/Event”.

- Proper design patterns **(1/2)**

You should not write business logics in controller. (-1)

- Normal level of refactoring in the code-base toward a better design, in every iteration. **(1/1)**
- a RESTful service (or any alterative) **(1/1)**

You should create separate business logic classes and use that from both mvc and api controllers.

- A missing API documentation (or missing info such as the location of the service and how to use it). **(0/-1)**

Your documentation of api usage was not enough. Remember, you must consume that.

- A missing installation guide for anything that is not in the documentation, like any special setting if required. **(0/-1)**

Give a little comprehensive instructions of how to run your Selenium tests. You should automate this.

Process and planning: **(2/3)**

- Team’s communication and members’ contributions **(1/2)**

You did not provide any meeting log (-1). You should always mention the link in the document.

- Following the plan **(1/1)**

Total **(18.5/25)**