**CSE 216 Project Manager Report: Phase 1, Team BMW**

Tanqiu Jiang: Project Manager

Ji Ho Choi: Android Developer

Allison Codi: Back-end Developer

Ryan Stelly: Web Front-end Developer

Karthick Sivakumar: Admin App Developer

**Back-End Server**

Describe the REST API endpoints that your team agreed to implement. If this deviates significantly from class discussion, explain why.

Our REST API endpoints stays almost the same as we discussed in class. We only added a few more explanations to the routes and deleted a few routes that we are not needed to create in phase1.

Describe the Data Model that your team agreed to implement. If this deviates significantly from class discussion, explain why.

Our data model has “ mTitle, mMessage, and mVote ”. It remains the same as we discussed in class.

Does the back-end match the description above? Why or why not?

Yes, all the routes were deployed successfully as we planned.

Is the back-end code appropriately organized into files / classes / packages?

Yes, every file of ours are well organized into files/classes/packages.

Are variables, functions, classes and packages named well?

Yes, we are keeping a documentation to keep track of all our variable/function/classes/package names.

Are the dependencies in the pom.xml file appropriate? Were there any unexpected dependencies added to the program?

Yes, the dependencies in the pom.xml file are set up in the same way as the tutorial.

What was the biggest issue that came up in code review of the back-end server?

The most challenge part is the route of upvote and downvote functions. The problems were solved after going to the office hour with TA.

What technical debt do you see in the current back-end server implementation?

The original plan was to create multiple tables for in the database. However, as the development progressing, we then decided to only create one table to get the program to work.

Are the unit tests for the back-end server complete/satisfactory? If yes, why do you have confidence in your answer. If not, what's missing?

There was an Async error on the unit test. Otherwise, everything else works fine.

What challenges did you observe on account of the relationship between the database and the server? Was mocking used to separate tests? Why or why not?

The biggest challenge was the upvote and downvote functions. Instead of using mocking, we developed our own code separately before the database was available, because in this way our progress won’t be slowed and we can still do the test when the database is finished.

**Web Front-End**

What is the overall structure of the classes used to implement the web front-end?

Singletons are used.

What benefits did you observe on account of the use of Singletons?

Instance control: Singleton prevents other objects from instantiating their own copies of the Singleton object, ensuring that all objects access the single instance.

Flexibility: Since the class controls the instantiation process, the class has the flexibility to change the instantiation process.

Does the user interface match the interface for the Android app? Why or why not?

The interface match the one for Android app for the most part.

Is the web front-end appropriately organized into files / classes / packages?

Yes, every file of ours are well organized into files/classes/packages.

Are variables, functions, classes and packages named well?

Yes, we are keeping a documentation to keep track of all our variable/function/classes/package names.

Are the dependencies in the package.json file appropriate? Were there any unexpected dependencies added to the program?

The dependencies in the package.json file are appropriate. There wasn’t any unexpected dependencies.

What was the biggest issue that came up in code review of the web front-end?

The edit button wasn’t functioning for a while but the issue was fixed later.

What technical debt do you see in the current web front-end implementation?

There hasn’t been a technical debt yet.

Are the unit tests for the web front-end complete/satisfactory? If yes, why do you have confidence in your answer. If not, what's missing?

Sufficient amount of Unit test has been done. The web developer is confident.

**Android App**

Are the mechanisms for moving among Activities satisfactory and appropriate?

Yes, all the mechanisms for moving among activities are according to the tutorial so we believe they are satisfactory.

Describe any significant deviations from the UI design discussed in class.

The UI design remains exactly the same as the one we discussed in class.

Is the android app appropriately organized into files / classes / packages?

Yes, every file of ours are well organized into files/classes/packages.

Are variables, functions, classes and packages named well?

Yes, we are keeping a documentation to keep track of all our variable/function/classes/package names.

Are the dependencies in the build.gradle file appropriate? Were there any unexpected dependencies added to the program?

Yes, the dependencies in the build.gradle file are appropriate.

What was the biggest issue that came up in code review of the Android app?

The hardest issue was to have the android app connected to the Heroku and fetch the correct data during the code review.

What technical debt do you see in the current Android app implementation?

There is still one minor issue to be solved: The UI will display the id of the message, which might confuse the users from the “like” counts. The future programmer might need to work on that issue.

Are the unit tests for the Android app complete/satisfactory? If yes, why do you have confidence in your answer. If not, what's missing?

Yes, the tests are satisfactory.

**The Admin App**

Is the Admin app appropriately organized into files / classes / packages?

Yes, every file of ours are well organized into files/classes/packages.

Are variables, functions, classes and packages named well?

Yes, we are keeping a documentation to keep track of all our variable/function/classes/package names.

Are the dependencies in the pom.xml file appropriate? Were there any unexpected dependencies added to the program?

Yes, all the dependencies in the pom.xml file appropriate. The admin app wasn’t functioning properly because of some discrepancies in the pom.xml file. The problem was resolved later in the code review.

What was the biggest issue that came up in code review of the Admin app?

The admin app wasn’t functioning properly because of some discrepancies in the pom.xml file. The problem was resolved later in the code review.

What technical debt do you see in the current Admin app implementation?

The original plan was to create multiple tables for in the database. However, as the development progressing, we then decided to only create one table to get the program to work. In the future we might need to go back to add new tables.

Are the unit tests for the Admin app complete/satisfactory? If yes, why do you have confidence in your answer. If not, what's missing?

Yes, the admin developer took a large amount of time on testing.

**Project-Wide**

Were any difficulties encountered when generating documentation for the Java and TypeScript code?

Yes, we sometimes face errors when we deploy new programs. However, the problems were solved eventually.

Were there any issues with the use of git for source control?

There were some overlaps between admin and backend in the Heroku tutorial. Therefore there was once a conflict on the backend branch. We then created another temporary branch for developing and deleted it when the projected is done.

Were there any team issues that arose?

There isn’t major issues during these two weeks since we are able to finish on time and still have one day left to write unit tests. However there actually were a few delays from our original plans on the admin.

How did the amount of time your teammates spent compare to the amount of time you thought the tasks would take?

Most part of the tasks were finished according to the plan we made. The only discrepancy was on the admin branch, which was delayed to finish by a week.

Describe the most significant obstacle or difficulty your team faced.

The most significant obstacle at this phase would be how to interact with the Heroku. We need to figure out how to deploy our programs on it, and how to connect to the database and fetch the information we need.

What is your biggest concern as you think ahead to the next phase of the project?

The biggest concern is that we might need to redeploy our programs on Heroku in the next phase to create more tables and add more functions. We need to figure out a way to still test our programs while the backend is still in progress.

Submit documentation for the user interface for the apps.

Submit the completed “CSE216 Project Manager Design Thinking Phase 1” form.