Team 32 Yelp Database Application Report

William Liu
Michael Sellean
Julian Tiu

CSCE 315-504
Dr. Sarker Tanzir Ahmed
24 October 2017

Evaluation of Team 31

1. Communication

At first, communication with the other team was slow. The first attempt of communication was due to the exchange of API library files. Due to an uncompilable API, we attempted to contact the other team, but received no reply until the beginning of the next week. However, towards the end of the project, the other team responded very well through text. Overall, Team 31 receives a 3 in Communication.

2. API Stage

The API did not stay the same for the duration of the time allotted for the project. The first library sent contained no documentation and was unable to be compiled due to linker errors. The linker errors were due to the essential class constructors not being defined. There were a lot of assumptions to be made especially because the other team utilized plenty of pointers to optimize the efficiency of their database design. An updated API was not sent to us until the end of the following week, and there was still very minimal documentation provided. Even by the end of the last week allotted for the project, there still needed to be alterations to get rid of runtime errors, although documentation got better towards the end. Overall, Team 31 receives a 2 in the API Stage.

3. Test Cases

The first API library sent was unable to compile, and there were no documentation provided. A proper test case was unable to be constructed, because there were a lot of assumptions made not accounted for by any documentation. As the other team was working on checkpoint 3 to fix their API, we received oral instructions on how to implement their design, yet there were still a lot of assumptions to be made in an attempt for a new test case. After checkpoint 3, the API design slightly changed, and only then was a proper test case constructed. The functions that failed were, min(), max(), size(), and makeKey(). Overall, Team 31 receives a 3 on the Test Cases.

4. Final Database

The Final Database still contains errors. This may be due to late updates on the test cases, so they were unable to further investigate the errors; however, the test cases were sent late, because a working API was not sent to us until the late stages. On the test cases, query worked fine; however, query errors were not found until the end of the implementation of the database application. As a result, the query does not work properly by the due date of checkpoint 4. Overall, Team 31 receives a 3 on the Final Database

5. Problem Handling

There were a lot of problems during the API stage and minimal communication. Throughout the majority of the time, the API would not compile correctly. However, towards the end, Team 31 made a lot of efforts to do as much problem handling once they had a working API. Overall, Team 31 receives a 2 for Problem Handling.

Team 29's Test Cases

We never received any test cases from Team 29. We also received very minimal communication from Team 29 with at most one email.

Team Evaluation Summary

In the beginning of the project, our team decided to split the workload by the number of classes – three classes with three members means one class per member. If one member finished early, then that member can help another member. The member that had the easiest class to implement also implemented the easier functions on the other classes, while the other members worked on the more difficult functions in the classes they were assigned to. The more difficult functions took longer, so if there were no more functions to be worked on by the member who finished the easier functions, then that member communicated with the other team to ensure minimal problems, misunderstandings, and miscommunications. This method proved to work as we met every deadline. The same method was used on making the database application for checkpoint 4, as the workload was split by three categories – parsing the JSON files and creating the database, interface that the program user can work with, and functionality of the database application. Given what we know now, we would keep the same method, as it proved to be effective in meeting the deadlines. Members can help each other out and take care of other jobs that would put fewer burdens on the ones working on the more difficult functions. However, a way to improve this method is to daily communicate updates made by each individual members, have more meetings, and organize time slots for each jobs and to make sure to meet these time slots on a regular basis. As far as problems and obstacles go, the most difficulty we encountered was communicating with the other team. The updates from the other team came pretty late, which made it difficult when it comes to the implementation of their API for the database application. In terms of grade, the team reached a consensus to give Julian Tiu a multiplier of .98, William Liu a multiplier of 1, and Michael Sallean a multiplier of 1.02, and these multipliers represent an accurate view of our relative contributions on the project that we all agree with.