

## **Evaluation of Team 17 Providing Library**

### **Communication**

Rating: 2

For every checkpoint that was necessary for us to have Team's 17 database, we always received it at least a day later than the actual due date. Their same excuse was that they were working on fixing minor bugs, which at first seemed reasonable, until we received their final database four days later. This caused delays for us in order to accomplish our final application, not to mention that we were not aware at how long they would take to finally give us their final database. Any questions we had about their database or confusion on its actual functionality was answered with a simply they knew it would not work, but they always advertised beforehand that their database passed every test case and it was therefore functional.

### **API Stage**

Rating: 2

The API provided documentation in order to understand how the code was structured, but it was changed over time and no further documentation was given in order to understand this aspect. It was a bit difficult to understand and it also did not provide the functionality that was necessary. As a result, many changes were made, such as making all the functions and structures public. Due to some of their functions not working, this unstructured code allowed us to actually implement our own functions to make the database properly work. Functions in the database such as simple querying, cross join, and natural join, did not work properly and were advertised as fully functional. For this reason, we replaced these simple functions with our own functions from our database in order to at least be able add functionality to our application.

### **Test Cases**

Rating: 5

The database was able to pass all test cases with no absolute issues despite being turned in a few days late. They gave us their database late in order to fix what they considered a minor bug. However, this meant we had to wait longer, but they wanted to make sure they passed all the test cases. We appreciated their desire to want to make sure their code worked properly and passing all their test cases was one of the few steps that was necessary. However, this gave us the impression that their functionality was working properly and it was advertised that way. This was in fact not true as they simply made sure their code could pass the specific test cases and not actually provide the full functionality.

### **Final Database**

Rating: 2

The database did not work as advertised. The final database was given to us four days after the date it was due because it did not compile initially. As a result, the other team communicated that they were trying to fix what they stated as a minor bug, but in fact did not let us know when they would be able to finish it in order for us to begin our application. We did not receive the final database until four days later with claims that their functionality was completely working properly. However, after only beginning minor usage of the database, we could quickly tell that this database was not fully functional as they had advertised. Their database did pass the tests cases, but it was specifically coded to pass only these test cases. It did not actually provide

the functionality that was necessary, such as basic querying. After experimenting with their database, we realized that the Query function does not handle CIN properly. Rather, the function utilizes the standard CIN operation without checking for errors. This results in errors in the query function being generated that are not being caught on standard use cases. Rather than catching the errors, it appears these errors were ignored by the coders. This works well enough for their implementation, however, once utilized the CIN errors propagate back to the calling program which is our application. This prevents us from using any form of standard input after the use of query from their database. Other aspects that were especially needed for the application, such as cross join and natural join, did not work either, which was essential. Therefore, this forced us to have to use some aspects of our database in order to achieve the basic needs of our application.

### **Problem Handling**

Rating: 2

The other team tried managing problems before we got the database. Problems such as compiling were resolved beforehand and they also tried resolving bugs they identified before we got the database. However, they didn't have any type of error handling in their database and they also promoted their database as fully functional. This was not the case and there were definitely parts of their code that did not function properly despite passing all the test cases. Instead of trying to fix this aspect of their database, they simply stated afterwards they knew about it but took no action to try to handle it nor did they try to improve it. As a result, after the database was given to us and advertised as properly functioning, they did not take responsibility for the actual functionality of it.

### **Different Team's Library**

We did end up using Team 17's database, but not entirely. As stated before, their functions were hardcoded to pass only the test cases they were provided, but did not actually provide functionality for any other aspect necessary. As a result, our team decided to use some of our functions in order to perform querying at least in order to apply it to our application. Some of these functions included cross join and natural join. This helped us in order to query the necessary information to display for our application.