

Project Reflection Report

Group: Ace_DB

- 1. Please list out changes in your project directions if the final project is different from your original proposal (based on your stage 1 proposal submission).**

Generally speaking, we believe that our project direction is consistent with the original proposal. We basically completed a board game application based on the review system, adding and modifying some functions to make it more in line with our goals.

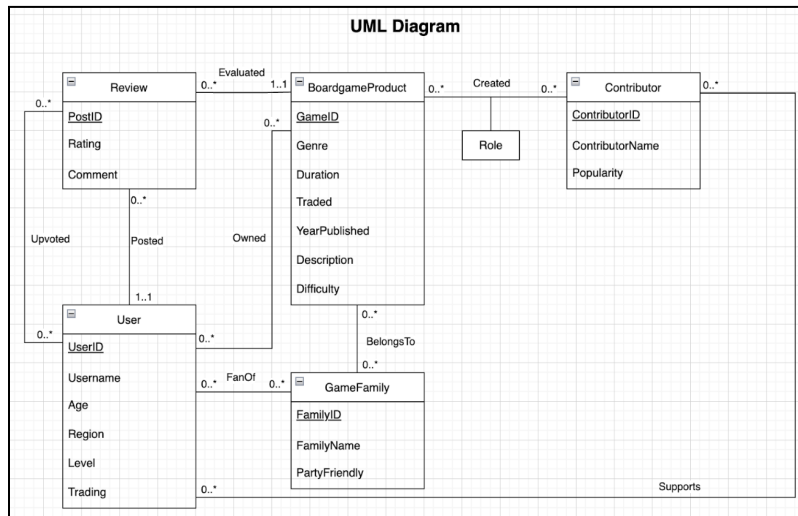
- 2. Discuss what you think your application achieved or failed to achieve regarding its usefulness.**

- **Achieved:** We implemented a basic board game application based on a review system. In this application, users learn about board games mainly through reviews and some additional information. Users can add, modify and delete reviews to keep them updated, thus ensuring the credibility and timeliness of board game reviews. In addition, the app also displays highly upvoted reviews from high-level users and profiles of games in the highest and lowest PartyFamily families to help users understand different aspects of board games. In addition, we also use the board game search function, users can search the board game according to the keywords, and learn the relevant information about the boardgames.
- **Failed to achieve:** Our application, while fulfilling most of its intended functions, is still a bit rudimentary and limited. For example, we don't have the functions to add users and boardgames, which means the app can only work with a limited number of users and boardgames. In addition, we do not use all the data in the database, and there is some data that we hardly use.

- 3. Discuss if you changed the schema or source of the data for your application**

The schema we designed did not change but we changed some sources of the data. We originally planned to use real data for all of our tables. However, the available data for users and contributors online did not meet our requirements. Thus, we generated some fake data for the users' demographic information, Contributor's popularity, and the specific user-upvote-reviews data was generated by us (since there was no available data from the matched source revealing the specific upvotes' details).

4. Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design? We did not change our UML. The original and final versions of UML are the same:



5. Discuss what functionalities you added or removed. Why?

- We removed the following functions:
 - For the search function, we have simplified the search bar so that users can only search for relevant games by game keywords.
 - We removed the comment search function. Because of the large volume of data in reviews, processing this data can take more time. For efficiency and utility reasons, we removed this function and added other functions to help users better understand this information.
 - For the Comparison Pane function, we adjusted it by comparing the boardgames in families with the highest and lowest Family to show players' preferences for board game families. Compared with the comparison between individual games, we believe that comparing the differences between board games in the unit of families can show players' preference for board game types from another perspective.
- We added the following functions:
 - Users can change their own information (age, region), so if we need to analyze the user distribution of board games later, this makes our data more realistic and timely.

-
- Users can add comments, modify and delete their own comments, which allows users to update their reviews at any time, making them more believable and valuable.
- Users can see reviews that have been reviewed by high-level players and those reviews have high upvotes, which allows users to learn which reviews are of high quality and learn about the board games from one perspective .
- Users can see the average number of upvotes and reviews for different levels of players in the app, which will help the users understand what his/her level is and what actions he/she can take to improve his/her level (the users' level will affect the result of the previous function).

6. Explain how you think your advanced database programs complement your application.

Our stored procedures complement our application by providing users the necessary information to engage with the board game community within the application. Stored procedure 1, "Title for Upvotes", allows users to view the average upvote threshold necessary to achieve a certain title in the community. Stored procedure 2, "Title for Reviews", functions similarly with respect to reviews necessary to achieve certain titles. Both of these stored procedures inform users, at-a-glance, the necessary engagement needed to achieve distinction within the community and web application.

Secondly, the trigger we have implemented complements the web application by providing a necessary level of incoming data quality control as it relates to users updating their information. The trigger prevents users who update their information with erroneous values to bypass making any real changes to the authoritative "User" table.

These will make our web application real and fun, because they added some real-world activeness statistics for goal-setting and the user-defined insertions with realistic considerations.

7. Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.

Andrea:

- *When we try to find tables which can be fully connected, it takes time in that we firstly need to look into potential available datasets online, and secondly we need to understand and match the data in data cleaning steps in order to prepare the data for our database.*
- *When we try to set up the front-end-back-end connection with GCP, it takes time because some of the tutorials were unfamiliar to me which means I need to first understand them before I can change it as a template for our use. The testing and set-up also takes time, because the local environments are different between different group members.*

Raina:

- *Echoing Andrea's sentiments, testing and setup was difficult due to different local environments. A recommendation I might make is to teach students how to implement their web applications directly through GCP.*

Tinghua:

- *Implementing different requirements at different stages was somewhat difficult, due to the need to constantly adjust front- and back-end code. In the future, possibly combining all steps required for stages 4 and 5 might be easier.*
- *Data preprocessing was lengthy, given the volume of our data. Additionally, some of our data had to be generated by the team. This proved to be a tedious task which took up considerable time in our early stages. In the future, giving students both a maximum and minimum requirement for records and/or tables would be useful.*

Xuecen:

- *Understanding and describing the performance in detail of our implemented indices was slightly difficult. While we had expected our defined indices to improve performance and optimize querying considerably, this was not the case, making it difficult for us to discern why.*

8. Are there other things that changed comparing the final application with the original proposal?

Front-end design and visualization could have been further polished with a greater amount of time, and more features could have been included with it. In previous stages, we had planned for a better user-interface for this system, implementing functionalities such as a comparison dashboard, in a visually

pleasing presentation. While the comparison function was realized, the dashboard was not implemented, unfortunately, although it would have been a beneficial addition.

9. Describe future work that you think, other than the interface, that the application can improve on?

Currently, there is no prevention of dirty read, unrepeatable read or write-write conflicts. In the future, an important prospective next step would include implementing a design to this system which prevents these types of concurrent executions.

Also, we believe that incorporating Neo4j would be possible since our data is highly connected. We have five relational tables, which is a bit tedious when running advanced queries. In Neo4j, the need to JOIN every table could be eliminated. More specifically and in example, in Neo4j, it would be helpful for us to store the User table and all related data to simplify and streamline retrieval from such advanced queries, since they are highly connected.

10. Describe the final division of labor and how well you managed teamwork.

In each stage, we divided the work dynamically because of unforeseeable obstacles and bugs we would encounter.

All of us were engaged in: brainstorming the idea of our app design and data sources, cleaning and manipulating the data for data preparation, UML design and writing the report.

Andrea: UI-design, set up the database and tables on GCP, developed two advanced queries and measured performance for one advanced SQL query, developed and realized Update and Delete of CRUD both back-end and front-end, realized two stored procedures and helped trigger development in stage 5, final demo and video presentation.

Raina: UI-design, data preprocessing, helped develop Update and Insert of CRUD, in charge of report editing and revision

Tinghua: set up the database and tables on GCP, developed and measured performance for one advanced SQL queries, realized Search, Insert, Update of CRUD both back-end and front-end, realized the display of two advanced SQL queries in front-end, developed and implemented the trigger and implemented of trigger and stored procedures both front-end and back-end in stage 5, mid-term

demo presentation.

Xuecen: UI-design, reviewed and examined two advanced SQL queries