

Final Report

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

The direction of our project does not deviate greatly from the original proposal, where we aim to build a web application to serve as a research paper database, allowing users to search for papers and save papers to read later.

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

The application is useful for

- searching for papers by keyword or title,
- showing the most relevant or highly rated results based on keywords, user ratings, and user activities,
- allowing user authentication, and
- allowing users to save papers, add notes, rate papers, and easily view information about saved papers after logging in.

Due to the limited time we had, the application failed to implement

- recommending users with papers relevant to their recent searches, and
- sorting the search results based on attributes (e.g. user ratings, published date) desired by the user as the search algorithm is implemented rigidly.

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

There are small changes for the schema. We have added `url` and `recent_add_date` attributes to the table Paper so that more details of papers are provided, and more relevant papers appear when searched. We have added an attribute-level constraint to the `rating` attribute in the table UserPaper, and removed the comment attribute from it because it is irrelevant to our implemented functionality.

There is no change to the source of data for our application.

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?

There is no change to our ER diagram for entities and relationships. The only changes made are the addition of attributes for two entities, as stated in Part 3. The attributes are added to support more complex functionalities of the application as the initial design was simple.

Discuss what functionalities you added or removed. Why?

We have added a functionality where the users can search for papers based on the rating category of the papers – high, medium, and low. This enables the users to filter good quality papers based on user ratings. In addition to that, the most recent added papers (by users) will be shown on top of the search results to reflect the current most relevant papers.

We have removed the annotation tool functionality because the application does not provide the paper itself, but an abstract and metadata of the paper. Therefore, the annotation is not very much necessary in this case. Providing the paper itself would have required creating a LONGTEXT variable representing each paper's contents inside the Paper table, which would have been difficult to store within the database.

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

The trigger is useful for updating two tables after an action. Specifically, in our implementation, after a paper is added by a user to their dashboard and this action is inserted into the UserPaper table, we get to update the Paper table with the most recent date the selected paper is added by a user. Also, in our application, the stored procedure and transaction are useful for the functionalities of our application such as filtering papers based on quality by user ratings. A stored procedure enables us to loop through the entries of the Paper table to update the average rating of papers and categorize the papers based on their ratings. A transaction is implemented to make sure to maintain concurrency control during the update.

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.

Yu Jun: Processing data and importing data into the database on Google Cloud was one of the technical challenges that I have encountered. I initially imported the data locally into a sqlite database but Google Cloud uses MySQL. Because of this discrepancy, I had to find a way to convert the sqlite dump file into a MySQL one. Eventually, I decided to import the data into the MySQL database using .csv files. However, it also took some time because of data cleaning and preprocessing.

Claire: One of the technical challenges I encountered was managing user sessions and authentication for the web application, which included tasks such as user login and account creation. I was unsure of how to load a user's saved papers on their dashboard because their ID was not being explicitly passed around after logging in. Node.js has an "express-session" module that creates a session middleware to simplify session management. I used this to store and retrieve user information as needed when implementing the POST/GET requests on the server side (for example, passing the user's ID from the search results of a GET request, to the "save" button on the client side, to the POST request to insert the paper into the UserPaper table with the corresponding userId and paperId).

Daniel: One of the challenges that I faced was when I was implementing filtering papers by keywords and similarity. I tried querying with multiple keywords and the lowest possible similarity value and expected results to populate, but to no avail. To investigate the cause, I checked if the Keyword and PaperKeyword tables were correctly populated, checked that the frontend was receiving responses, but still couldn't figure out. To my surprise, it was a very simple problem. Because a single paper could have many keywords and I intended to add the functionality of querying by multiple keywords, I was running queries of the form `SELECT DISTINCT paperId...`, where the papers were ordered by date added, as the other paper queries were. But I forgot the fact that you can't use `SELECT DISTINCT` and `ORDER BY` a field that isn't being selected.

Ryan: One of the challenges I encountered was managing the modals to be accessible and usable with their buttons. There were multiple instances of modal buttons not working when opening a new modal to insert data. I diagnosed this by moving those buttons outside of their modals, and discovered the issues behind them there. They had been of the wrong class, causing the scripts that they were calling in the rest of the code to be called on the wrong buttons without proper input. They were trying to parse input that did not exist, causing a failed query and an empty return. Once I changed which function the button was supposed to call and modified the button to be what it needed to be, the issue resolved.

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

We managed to implement all the basic functionalities of the application stated in the original proposal, but did not implement the recommendation system and annotation tool. There are slight changes between the current user interface and the UI mockup stated in the proposal, but those do not contribute to important feature changes for user experience.

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

Some future work that can be done to improve the application are as follows:

- Improve the speed of database retrieval (through experimenting with indexing, creating more stored procedures, avoiding unnecessary data retrievals in queries, etc.)
- Add more filtering options so that the search results are sorted based on attributes (e.g. user ratings, published date) desired by the user
- Expand the database by including more papers from different fields apart from computer science and technology
- Implement a recommendation system
- Enable 'Add friends' and 'Create/Join Group' features to share papers among a group
- Enhancing security (adding authentication methods/detecting suspicious behavior, incorporating input validation techniques to protect against SQL injections, etc.)

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

Yu Jun: Database design and normalization, keyword extraction, database creation, data preprocessing, SQL queries, backend

Claire: tf-idf implementation, frontend/backend connection (express-session, ajax, get/post requests), user functionality (login, create/delete account, save/remove paper, add rating/notes), search functionality (filter by rating)

Daniel: frontend/backend connection (express-session, ajax, get/post requests), search functionality (filter paper by keyword and keyword score), SQL queries

Ryan: Backend SQL server and VM construction and maintenance, frontend/backend connection (express-session, ajax, get/post requests), UI/UX CSS, HTML, SQL queries