Sprint 4 Report

🥐Team Scrum-ptious🥐

What functionality does the system have at the end of this sprint? List user stories that you successfully implemented during this sprint (5 pts)

16. As a developer, I want to document my knowledge so that people who inherit my code can understand the code structure and my decisions.

- Size: 13
- Priority: High
- Precondition: We have implemented all planned features and have access to prior research
- Postcondition: Future developers will be able to interpret our work and not repeat the research/implementation we have accomplished

17. As a developer, I want my endpoints to accommodate the most current schema design to ensure data consistency.

- Size: 5
- Priority: High
- Precondition: We have implemented an endpoint which does follow updated schema requirements
- Postcondition: All endpoints return responses which are expected by agreed upon schemas

18. As a developer, I want to consolidate my work along with my teammates in a single branch and make a pull request to main.

- Size: 5
- Priority: VERY High
- Precondition: Team has all of their changes in separate branches
- Postcondition: Team has submitted all their changes in a pull request under one branch

19. As a developer, I want repetitive API calls to be stored in a cache in order to reduce fetch time from Wikipedia.

- Size: 8
- Priority: Medium
- Precondition: The endpoint which will fetch data from Wikipedia is in place

- Postcondition: Repetitive calls will be stored in a cache temporarily, reducing unnecessary calls to Wikipedia

Key functionalities Implemented:

User Story 16: We have written a very detailed README.md for the next team that is handed off our work in the summer in order to seamlessly be able to onboard. It documents installation details, current functionality, research, and next steps; it is very similar to what is present in the functionality guide. We also created multiple slide demos for final functionality which will be available to the next team to have some visual explanations of our work.

User Story 17: Our current comparison schema now returns data in the form outlined by the most recent schema we agreed on with the other teams. However, as the ML team's current code has not been pulled into the main branch and does not support the latest schema, we are returning dummy data for the moment. Connecting live data is left to future implementers.

User Story 18: We managed to successfully merge our branches. This involved resolving multiple conflicts across a handful of files. We still have to get approval for our pull request to the main branch, but are otherwise done with this task.

User story 19: As a bit of a last second task born from passion, we have implemented a caching feature which will store both the most recently fetched article (and its languages) into a cache so that if repetitive queries are made referencing this data, more requests will not be sent to Wikipedia, heavily reducing fetch expenses. The cache was initially implemented with a memory leak (the cache only flushed outdated articles when they specifically were queried), which has been repaired.

Did you end up making any changes to any of these user stories? Did you break down any further user stories? Did you identify any new user stories during this sprint, and if so, did you add them to the product backlog or decide to implement them right away? Explain (5 pts)

NEW STORIES:

19. As a developer, I want repetitive API calls to be stored in a cache in order to reduce fetch time from Wikipedia.

- Size: 8
- Priority: Medium
- Precondition: The endpoint which will fetch data from Wikipedia is in place
- Postcondition: Repetitive calls will be stored in a cache temporarily, reducing unnecessary calls to Wikipedia

What are the "lessons learned" at the end of this sprint? What would you do differently next time? Explain (5 pts)

- Documentation - We learned that creating efficient documentation is a great way to reflect on the work that you have done, and a great way to find out what to do next. Upon creating our README and doing a comprehensive code review, we realized that we had missed a few problems in code that we had inherited that reportedly worked. We have documented the shortcomings for the next team so that down the line the issues don't cause them headaches. Additionally, some members of our team learned how to use Markdown in order to express information neatly.

- Reasonable Workload - For this sprint, it is pretty obvious we took on a lot. We were able to get everything completed for our sprint, but are a bit behind on our offboarding with Grey-box (video demos still need to be made, profiles need to be updated, etc). In the future, we will take into account time constraints more reasonably (one week sprint as opposed to two) and be sure to set realistic goals for ourselves, especially for the last sprint.

- Task Prioritization - There were some features that we decided to implement relatively last second which are very cool, but made it more difficult to wrap up our work. We tried to figure out how to implement caching in the tail end of our third sprint, but discovered the memory leak in the implementation during review. As a result, we temporarily removed it from our combined branch and have delayed our final merge to ensure the updated version makes it into our completed work.

Provide an updated numbered list of all user stories yet to be implemented; indicate pre- and post-conditions (5 pts)

Remaining User Stories:

NONE!

Are there any user stories left unimplemented in the backlog? Are there any new user stories that you would consider adding to the backlog. List these user stories and explain them. (5 pts)

20. As a Wikipedia user, I want Symmetry to fetch the correct article in another language despite conflicting titles.

- Size: 8
- Priority: Medium
- Precondition: There is an article in a language whose name in the first language is a valid page in the second language (but not the page we are looking to compare with)
    - Ex: https://en.wikipedia.org/wiki/Gift and https://de.wikipedia.org/wiki/Gift - One is a gift, one is poison! https://de.wikipedia.org/wiki/Geschenk would be the correct page!
- Postcondition: The correct articles are compared

Note: This user story was discovered when we considered the fact that titles of articles in alternate languages may be ambiguous or conflicting. In such cases, Wikipedia may not redirect to the correct article. This is not considered in the current implementation.

21. As a developer, I want terminology regarding schemas to be consistent so that communication about data responses is clear.

- Size: 3
- Priority: High
- Precondition: All schemas are in place, with some inconsistent terminology being used to describe them

- Postcondition: All schemas follow the same key naming conventions so that there is no confusion what data is being returned and requested.

Note: As of right now, we have some slight naming inconsistencies in our schemas that we are using (“text_blob” vs “left_article”, etc.) which might cause confusions regarding terminology and data being handled in the future.