

CSC302 A2

Zixiao Ren, Qiming(Austin) Ye

Feature Description and Prioritization

Note: The feature assigned with a lower number has a higher priority.

Features plan to be completed in A2

Home page:

1. Search for channel by channel ID
2. Display search results
 - a. Channel info: channel title, description, published at, custom url, followers count, playlist count
 - b. Playlists info: A redirection button to the playlist info page
 - c. Videos info: A redirection button to the video info page
 - d. Back: A redirection button to the home page
3. Playlists Info Page
 - a. A table to display playlists info: playlist title, description, published at, video number, total # of likes and view count
 - b. A redirection button to go back to the channel page
4. Videos Info Page:
 - a. A table to display videos info: video title, description, view count, like count, actual start time, scheduled start time(if available), duration
 - b. A redirection button to go back to the channel page

Tasks:

- Feature 1
 - Due date: Nov 2nd, 2022
 - Acceptance criteria: The user can enter a channel ID into the search bar. After the user clicks on the search button, the page should either be redirected to the channel info page if the channel is found. Otherwise, it should show the user a channel not found error.
 - Zixiao Ren: works on the search bar frontend
 - Qiming(Austin): works on the backend API to get related information
- Feature 2
 - Due date: Nov 5th, 2022
 - Acceptance criteria: It should show the user the info of the searched channel and buttons for page redirection. (Note that the buttons should be able to redirect the page to the playlist info page and videos info page at this step.)

- Qiming(Austin): Frontend (no backend needed, the backend API that is used to get playlist info has been done in Feature 1.) and test script for channel info backend API
- Feature 3
 - Due date: Nov 5th, 2022
 - Acceptance criteria: It should show the user playlist info of the searched channel and a button to redirect back to the channel page.
 - Zixiao Ren: Frontend (no backend needed, the backend API that is used to get playlist info has been done in Feature 1.) and test script for playlist info backend API
- Feature 4
 - Due date: Nov 5th, 2022
 - Acceptance criteria: It should show the user video info of the searched channel and a button to redirect back to the channel page.
 - Qiming(Austin): Frontend (no backend needed, the backend API that is used to get playlist info has been done in Feature 1.) and test script for video info backend API

Further features for A3

1. Video title keyword recommendation page: Find a recommended list of keywords for video title (based on the keywords with high frequency in the title of previous popular videos)
 - a. A redirection button on the channel page for the redirection to this page
 - b. A drop-down menu for the following choices
 - i. The recommended keywords based on all the videos
 - ii. The recommended keywords based on videos in a single playlist
2. Video description keyword recommendation page: Find a recommended list of keywords in the description of popular videos (based on the keywords with high frequency in the description of previous popular videos)
 - a. A redirection button on the channel page for the redirection to this page
 - b. A drop-down menu for the following choices
 - i. The recommended keywords based on all the videos
 - ii. The recommended keywords based on videos in a single playlist

Tasks:

- Feature 1
 - Due date: Nov 18th, 2022
 - Acceptance criteria: It should have a drop-down menu that allows the user to choose how the recommended keywords for the video title are found. After the user makes a choice, it should show the user a list of keywords based on the user's choice. It should also have a button to redirect back to the channel page.
 - Both: Design algorithm for finding recommended keyword

- Zixiao Ren: The recommended keywords based on videos in a single playlist
 - Frontend and backend implementation
- Qiming(Austin): The recommended keywords based on all the videos
 - Frontend and backend implementation
- Feature 2
 - Due date: Nov 26th, 2022
 - Acceptance criteria: It should have a drop-down menu that allows the user to choose how the recommended keywords for the video description are found. After the user makes a choice, it should show the user a list of keywords based on the user's choice. It should also have a button to redirect back to the channel page.
 - Zixiao Ren: The recommended keywords based on videos in a single playlist
 - Frontend and backend implementation
 - Qiming(Austin): The recommended keywords based on all the videos
 - Frontend and backend implementation

Note that there might be no testing for the features in A3 since it is impossible to test the accuracy of recommendation algorithms. Also, these features would only need to call the backend APIs that have been implemented and tested in A2. (However, if later in A3 we find there is anything that needs to be tested or if we have time to figure out how to test the recommendation algorithm, we will include the automated testing in A3 tasks)

Progress Demonstration

In A1, we designed 4 milestones for this project. However, later we found that we only have A2 and A3 left in this course. In order to evenly separate the workload into A2 and A3, we decided to complete Milestone 1 in A2, Milestones 2 and 3 in A3, and Milestone 4(testing) in both A2 and A3. The reason we separate the milestones in this way is that it is always hard at the beginning of a project, so it is good enough to complete only one milestone in A2. The features in Milestones 2 and 3 are very similar, so these two milestones should be started at the same time. Moreover, since it is always a good practice to do the testing whenever we complete a feature, both A2 and A3 should work on Milestone 4. Note that only the testing for the features in Milestone 1 can be automated. The testing needs to be done in just one command line, it is impossible to do both frontend and backend testing in just one command line, so we decided to manually test the frontend and automate the backend testing. Since all the backend APIs are implemented in Milestone1(A2), we only need to work on the automated testing in A2. (However, if later in A3 we find there is anything that needs to be tested or if we have time to figure out how to test the recommendation algorithm, we will include the automated testing in A3 tasks).

In A2, we need to finish the features in the “Features plan to be completed in A2” section above. In order to make sure we finish our tasks on time and all the features are implemented in the

way we expect, we had a meeting to discuss details about the frontend UI design for all the features, feature prioritization, subtasks for each feature, and the deadline for each task. All the details are written in the meeting notes and the “Feature Description and Prioritization” section in this document. To ensure everyone is on the same page during the whole development progress, we had a meeting to update everyone’s progress and discuss more detail about the frontend coding structure and the optimization of the frontend UI design(more details can be found in the second meeting notes). The last meeting that we had was to wrap up and see if there are any uncompleted tasks that need to be done before the deadline.

Up until Nov 4th, most of the subtasks are completed. The user can search for a channel by the channel ID and all the channel, playlists, and video information can be shown to the user as expected. The user can also redirect the pages by clicking on the buttons. However, since both teammates have heavy workloads in the week of Nov 4th(Zixiao had 6 assignments due this week and Qiming had 10 assignments to work on this week), we did not have time to push the code to the cloud and work on testing and bug fixing. All the uncompleted tasks should be finished before Nov 6th. By the end of this assignment, for the validation progress, we need to check if the milestone has been achieved. To check this, we need to manually test the frontend based on the acceptance criteria that we set before and run the automated tests for the backend. If we pass the acceptance criteria and all the backend testing, then this means the milestone for A2 has been achieved.

Since we found that although we set the deadlines for all the subtasks, it is still possible that we cannot complete the tasks on time, for the next milestones in A3, we need to set the deadlines based on each other’s calendar so that we can be clear on when do we have time for this assignment. We can also set up meetings to check the status of each task. For the completed tasks, we can check if they met the acceptance criteria on the frontend before testing the backend so that we can avoid the situation where the backend logic needs to be changed for the frontend after the backend test scripts have been completed.

Current State

GitHub: <https://github.com/Austin618/CSC302-Assignments/tree/main>

Command to run the dev environment: `docker-compose up` (then open localhost:5000 to view the website)

Command to run the testing: `docker-compose -p tests run csc302assignments npm run test`

Blameless Postmortem on Assignment 1

- What goals have been achieved.
 - Implementations of all the features in Milestone1.

- Automated backend testing and frontend manual testing for the features in Mileston1.
- What goals have been missed, and by how much.
 - None. All the goals for Assignment2 have been achieved.
- How plans will need to be adjusted, based on remaining time available and project scope.
 - In A1, we set 4 milestones and one milestone for one assignment. However, we found that we only have two assignments left in this course, so we changed our plan to be completing Milestone1 in A2, Milestones 2 and 3 in A3, and separate Milestone 4 into A2 and A3 (more detail can be found in the first paragraph of the Progress Demonstration section).
 - There were some “good to add” features we designed in A1. These are the features that are good to have if we have extra time to implement them. However, based on our heavy workloads in November, we don’t think they are possible to be implemented, so we decide to not implement them for our project.