

Assignment T2: Revised Project Proposal

Team members:

Ping-Feng Lin, pl2730

Yuan-Hsi Lai, yl4305

Yen-Min Hsu, yh3328

Shao-Chi Wu, sw3525

Team name: TaiOne

GitHub repository <https://github.com/Jefflin413/ASE-team-project>

IA: Asif Mallik

Meeting with IA: 2020 NOV 15 13:00

Part 1 Description of the software engineering project, answer the following questions

1. What will your project do?

This project presents a live streaming platform for streamers and audiences, as well as companies to advertise. We currently only work on taking personal video recording devices as input, i.e., webcam and phone camera. When a registered member wants to start streaming, depending on whether the exclusive channel pipeline exists, we either trigger the construction of AWS live streaming solution and connect the member's device with the resulting channel or turn on the existing streaming pipeline to start streaming. Audiences can access live shows from the "current streaming list" shown on the home page. By clicking on one of the links from the list, the audience will be redirected to another page which will have a video player showing the live streaming.

2. Who or what will be its users? (registration, authenticated login and timeout/explicit logout)

The first user role would be the streamers, people who are excited about showing their life/perform/play to the public. The second user role would be the audiences, people who enjoy watching and immersing themselves in the atmosphere of having fun together. The streamers and users are both members of the platform, and their roles are interchangeable. The third role would be companies, or to be specific, corporations who want to put advertisements on the platform. These companies will need statistical data such as the number of registered members, active users per hour, the most popular show category or streamer and so on.

We will be using Google Authentication through OAuth and Flask-Login extensions for Flask to perform authentication and registration. This would prevent leakage of user passwords as we don't actually store the login credentials. Either user from the above

three roles will have to register first though /register webpage and /login every time they want to access the platform.

3. project must be demoable (All demos must be entirely online)

We can demonstrate a full process of starting with a non-registered guest to becoming a live streamer on the platform. Also, we can be the audiences as we click one of the links on the list shown on the homepage. The demonstration for companies comes in two parts, one is that we can show the retrieved statistical data as a reference for companies who might want to invest their money into the platform. We can generate a report for them and can also do a match with the companies' requirements. The other one is we will show advertisements on the audiences' interface in the form of text or picture, and the type will be decided according to their watching histories.

4. store and retrieve some structured application data persistently, What kind of data do you plan to store?

User's personal data (account, email, personal information), watch histories, the exclusive streaming pipeline authentication (probably a secret key or any required element for identifying for connecting to specific streaming pipeline), statistical data.

5. Your project must leverage some publicly available API beyond those that "come with" the platform (external data retrieval, library or service)

AWS SDK for python

We use the API for CloudFormation to construct and integrate multiple AWS services, roles(policy) and communications which are necessary for supporting the live streaming function. CloudFormation is a service that can deploy a predefined template to be a complete technology stack to produce the desired service. The template we are using contains MediaLive, MediaStore, S3, Lambda and CloudFront. By using this API, we can check the status of the stack, control it, access to each independent service and retrieve wanted data, for example, we can get the m3u8 URL that is needed to be put into the website embedded video player to display the live. We can also get the endpoint that the streamers need to transmit their content to. Thus make the live streaming possible. We can also obtain the performance data from Cloudwatch while it is always monitoring the operation of the whole system.

Part 2 Write three to five user stories constituting a Minimal Viable Product (MVP). Use the format

s00: As a **live streamer**, I want to show my exciting life in a cool way to my subscribers/audience so that I can be popular and probably earn some money.

My conditions of satisfaction are:

1. I can sign up with my own account.
2. There should be a web page showing relative information, e.g. how many people are watching, about this streaming when I am using OBS to connect to the system and perform my streaming.
3. There should be an option to turn off the advertisements.
4. [Stretch] I want to know the feedback from my viewer so that I can adjust what I'm going to do or respond to their comments in my streaming, creating a sense of interaction.
5. [Stretch] I must be able to save/keep the comments of my viewers.

error/exception:

1. If the streamer terminates the OBS accidentally , all the viewers get a message indicating that it's a streamer side problem.

v00: As a **subscriber**, I want to watch quality streaming so that I will feel a little bit better about my miserable life.

My conditions of satisfaction are:

1. I want to sign up without much effort.
2. I want to receive notifications of my favorite streamers' activities so that I won't miss them and be sad.
3. [Stretch] I want to know how much time in total I have spent watching.
4. [Stretch] I want to set the quality of streaming video received so I can still watch while I am at a relatively low downloading speed.
5. [Stretch] I want to know what my friends are watching.

error/exception:

1. If a subscriber uses a browser our system does not support such as IE, there should be a message saying that "your browser does not support this, please change to firefox."
2. If a subscriber wants to leave, there should be a pop-up window showing for this subscriber to double check.

c00: As a **corporation**, I want to put advertisements on the platform so that our image advertising strategy can work.

My conditions of satisfaction are:

1. I will need statistics such as the number of registered members, acting users per hour, the most popular show category or streamer and so on. To be more specific, I need all the analytics of all the streamers.
2. This system should support advertising in the form of images
3. I want to know how many people have watch our advertisements
4. I want to let people with different languages watch the corresponding version of our advertisement.

5. I am capable of assigning ads to specific streamers who I think will maximize our company reputation.

error/exception:

1. When a corporation tries to assign advertisements to a live streamer whose advertisement function is turned off, this operation won't be valid and the corporation user gets a message from our system.

Part 3 Explain how you will conduct acceptance testing on your project.

Basic Flow Acceptance Testing

1. Player 1, representing a streamer, gets a key and an url generated by the system. Player 1 enters them into OBS and opens the camera to start streaming.
2. Player 2, representing a viewer, connects to the website and uses an account to login. Player 2 clicks on one of the streamers (not player 1) shown on a customized homepage and finds that the streaming is not started yet. Player 2 goes back to the home page and clicks on the channel of player 1 and player 2 can watch the live stream.
3. Player 1 ends the streaming normally and player 2 gets proper notification.

Acceptance Testing with Mock Database

1. Player 3, representing a corporation's marketing staff, logs in to our system and reads the reports generated by the system. Then, Player 3 clicks on the advertisement selection page to assign company advertisement to Player 1 who is a streamer. Player 1 started streaming as Basic Flow Acceptance Testing 1.
2. Player 2, representing a viewer, connects to the website and uses a google account to login. Player 2 clicks on the channel of player 1. Player 2 can watch the live stream with an advertisement of Player 3's company in the corner.

Error / Exception:

1. Player 1 who is a streamer turns off the WI-FI to terminate OBS, which is not a normal way to end a live streaming, and Player 2, representing a viewer, gets information about what's happening. (successful ended, streamer side problem, client side problem).
2. Player 3, representing a corporation's marketing staff, uploads an Excel file or an image that is too large to put on the page, and the system pops error messages that indicates file format is wrong or that the size exceeds the limit.

Part 4 Identify the specific facilities corresponding to JDK, Eclipse, Maven, CheckStyle, JUnit, Emma, Spotbugs, and SQLite that your team plans to use.

Compiler/runtime (or equivalent): Cpython

An IDE or code editor: Spyder, Pycharm

A build tool (package manager): Anaconda

A style checker: Pycharm and Spyder has built in style check, or could use pycodestyle 2.6.0 (<https://pypi.org/project/pycodestyle/>), a python package for checkstyle

A unit testing tool: unittest package, should be included in Anaconda already

A coverage tracking tool: Coverage.py (<https://coverage.readthedocs.io/en/coverage-5.3/>), a python package to test coverage.

A bug finder: PyChecker (<http://pychecker.sourceforge.net/>)

A persistent data store: SQLite