Assignment T1: Preliminary 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

1. **Description of the software engineering project, answer the following questions**

1. What will your project do?
A: 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)
A: 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.
3. project must be demoable (All demos must be entirely online)
A: 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?
A: User's personal data (account, email, hashed password, 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)
A: AWS SDK for python, Django or Flask,…

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

< label >: As a < type of user >, I want < some goal > so that < some reason >. My conditions of satisfaction are < list of common cases and special cases that must work >.

`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 want to know the instant feedback from my vierwe so that I can adjust what I'm going to do or respond to their comments in my streaming, creating a sense of interaction.
2. I must be able to save/keep the comments of my viewer
3. There should be an option to turn off the advertisements

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

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

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.
2. I want to know how many people have watch our advertisements
3. I want to let people with different languages watch the corresponding version of our advertisement.

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

1. An audience can view the entire streaming without interruption or additional action (like switching rooms, reconnect, etc.), and the streaming should provide quality above 480p if the Internet is sufficient.
2. When an audience is entering our website, all the available live streams should be shown on the selection page, and when a live stream expires, it should be removed on the page.
3. An audience is able to show his/her love to the streamer, and the streamer received real time.
4. The streamer can stream video at least 480p for an hour (if the Internet is sufficient) without any interruption or additional action (like switching rooms, reconnect, etc.), and he/she can also respond to the viewer's reaction real time.
5. The company gets the number of active viewers real-time and can put their logo on the home page by request.
6. Any ads are not blocking any important information on our site.

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