The Snake Squad -- Yifei Wang, Yuanxin Yang, Harsh Yadav, Daria Manea, Ki Cooley-Winters

Part 0:

Daria and Harsh have been added to the group since there are 3 members in the group initially.

Part 1: **-Daria-**

1. What will your project do?

Our project will implement a productivity timer that is based on the pomodoro technique. The timer can be started by the user and after 25 minutes of work, the timer will stop and automatically start a five minute timer which represents the break. The cycle will continue for as long as the user wants. For each user's account, information about the productivity intervals is stored so the user can review statistics about the productivity intervals that they completed in the past day/week/month.

2. Who or what will be its users? Your project must impose registration, authenticated login and timeout/explicit logout, so your answer should say something about this.

The users will be college students or more generally, people who are interested in the idea of the pomodoro technique. Each user can log in with their email in order to keep track of their progress. At the restart of the browser, the user must login again.

3. Your project must be demoable, but does not need a GUI if there's a command line console or some other way to demonstrate. (All demos must be entirely online, there will be no in-person demos.) What do you think you'll be able to show in your demo?

During the demo, we should be able to demonstrate how the timer looks like while it is working, how the notifications work when the productive interval is over, and when the break interval is over. We also could demo viewing the statistics after the user stops the timer.

4. Your project must store and retrieve some application data persistently (e.g., using a database or key-value store, not just a file), such that when your application terminates and starts up again some arbitrary time later, the data is still there and used for later processing. What kind of data do you plan to store?

We are planning to store the interval information for each user. The user can later view statistics about their progress such as how many productive intervals they have done for a given day, week, month.

5. Your project must leverage some publicly available API beyond those that "come with" the platform; it is acceptable to use an API for external data retrieval instead of to call a library or service.

(Links to an external site.)

and https://github.com/n0shake/Public-APIs

(Links to an external site.)

What API do you plan to use and what will you use it for?

For the login and authentication processes we plan to use Django:
https://www.django-rest-framework.org/api-guide/authentication/

For the web classification, we plan to use https://www.webshrinker.com/apis/

For the database, we plan to use mysql
https://dev.mysql.com/doc/index-connectors.html

Part 2: Yuanxin Yang

1. As a college student who is preparing for finals, I want to manage my time so that I can study more efficiently and more thoroughly.
My condition of satisfactions:
a. keep tracks of how long I have been study and schedules break time wisely over the final week
b. I should be able to review how much I have done

2. As a housewife, I want to schedule my day so that I can be more organized about housekeeping and can take care of the family better.
My condition of satisfactions:
a. the timer should help dividing up my day into chunks with tasks as labels
b. time length of each portion should be able to be adjusted

3. As a violinist, I want to spend some time every time to practice so that I can improve my skills in playing violin
My condition of satisfactions:
a. Depending on how busy I am on that day, I can adjust my practice time
b. The timer should tell me how concentrate I am in practising, for example, measuring the “silent period” and reports it

4. As a software developer who sits all day working, I want to spend some time doing some workouts everyday so that I can live a healthy lifestyle.

My condition of satisfactions:

a. As a beginner, I want to make a not-so-intense workout cycles
b. I want to be able to see how many hours of exercise I have done over some period of time so that I can keep a reasonable amount of workout while not over exercising

5. As an entrepreneur who has a very tight schedule everyday, I want to utilize my fractional spare time(for example the time to go to work, go back home, after lunch break, etc) not only for relaxation but also for study so that I can improve myself and do a better job at managing the company.

My condition of satisfactions:

a. Since such spare time is precious to me, I have to know the study efficiency time interval of myself to decide on which spare time period as my study time.
b. Since all my study periods are within my fractional spare time, the study timer should make the study interval short but frequent so that I can utilize my spare time better.

Part 3: Ki

1.
    a. When the user begins a timed session, the timer should start and display the amount of time left in real time. Otherwise, the test should fail. When a study session ends the user should be notified and the study session should be saved. Otherwise, the test should fail.
    b. If a user pauses a timed session before it has ended, the timer should stop and the partial session logged. If a full session or no session is logged, the test should fail. If the session is unpaused by the user, the timer should begin again at the same point. If it restarts instead, the test should fail.
    c. If the user closes their browser or the application crashes for any other reason, the previous session should not be logged and the timer should be reset upon restarting the timer. If the interrupted session was logged or the timer is not reset, the test should fail.
2.
    a. When the user starts the application and sets the timer to a certain value, the timer should begin at that value when a session is started. Otherwise, the test should fail.
    b. If the user starts a session without specifying an interval, the timer should start at the default of 25 minutes. Otherwise, the test should fail.

c. If the user tries to set a value less than 1 second or more than 2 hours for the timer, the timer should be set to the default value and the user informed that those values are invalid. If the timer is set, the test should fail.
d. The user should be able to specify a label when configuring a session. Any valid text string less than 60 characters should be accepted. If an invalid string is accepted, the test should fail. The label should be logged with the session, otherwise the test should fail.

3.
a. When the user has previously completed a session, the information (length, start time and date, label) should be displayed alongside other previous sessions. If the saved session info is not visible, the test should fail.
b. If the user has not completed any sessions, there should not be any information available, otherwise the test should fail.

4.
a. When the user has previously completed at least one session, the aggregate amount of time logged over some period (day, week, month, as specified by the user) should be visible, otherwise the test should fail.
b. If the user has not logged multiple sessions with the same label, the aggregate view should only show the single session. Otherwise, the test should fail.

5.
a. When the user has logged multiple sessions of similar lengths (grouped within +- 15 minutes of each other), the user should be able to see the number of sessions logged within various length groups, otherwise the test should fail.
b. When the user specifies a break interval and number of repetitions and then starts a session and it ends, the timer should display and count down the break time and then restart a session of the same length. If the timer does not restart after the break or the break session is logged, the test should fail.

Part 4: - Harsh

Language: Python

Framework: Django

Hosting Platform: AWS EC2

Runtime: PyCharm

Code Coverage Tracker: Coverage

Unit Testing Tool: doctests, nose, and attest

Style Checker: Code Style

Bug Finder: PyChecker

Data Store: MySql, DynamoDB

Submission instructions: One member of your team should submit a single file (preferably pdf) presenting your preliminary proposal. Your file must contain your team name and the names/unis of all team members, and may optionally include links to external resources. You may submit repeatedly until the deadline (note that if multiple team members submit, the most recent submission will override all previous submissions by the same or different team members).