Join Me: An Event Management Platform

Second Iteration Demo

Team: MissingOne
Jiayi Li(jl4924), Chi Zhang(cz2465), Chengyun Yu(cy2468)

1. The date and time at which you already completed this demo, and briefly describe any challenges that arose before or during the demo.

We completed the coding of the demo on Nov. 27th and demonstrated it to our mentor on Nov. 29nd. During the demonstration of our demo, the process was pretty smooth. We didn't meet much challenges but only a tiny compatibility issue. The front end was developed on OS X but we demonstrated the demo on Windows. As a result, the UI design of our desktop application looks different on the two different OS. But it is not a big issue, since when we can solve it by transferring the project files to the exe. executable file. During the development process, the selection of GUI framework in Python is actually a big challenge, since, to be honest, there's not much developer will use Python to do the frontend developing work while Python is really powerful in backend work. We compared many different GUI framework and finally chose the PyQt as our GUI framework for its UI design can be really nice compared to the other framework.

2. The specific use cases that were demonstrated, highlighting any changes since the First Iteration Demo.

We have demonstrated two user cases during the demo session. The first case we demonstrated was event host posting and managing events within the system. First, we logged in to a pre-established account, and posted an event under that account. We were able to save the descriptions, location information and links to descriptive images in our database. We then edited the details of the event, and verified the changes in the database. Also, through screen display from our application, we observed the same contents within the page that displays event and they were displayed as intended. Finally, we logged into another user's account, and was able to find the same event through search.

The second case we demonstrated is event seekers finding and joining events. Continuing from the demo of the first case. After successfully logging in, we were able to display the previously hosted event on the search result list based on its common category and location with the user's profile. By clicking on the event, the application jumps to a page where the details of the description were shown as posted before. By clicking on the “Join” button, the event is moved to the joined event list of the user. We can then again view the event from the list.

Key changes from the first iteration includes:

1. Mechanisms to log in or sign up with Google account.
2. The ability to handle image uploading and display.
3. Separation of server and client code
4. User interface with multiple windows and redirection logic.

3. The specific CI mechanisms that were shown during the demo, including which technology you used.

In the CI part, we used the suggested CI tool, ***Travis ci*** to do unit test and branch coverage test for post-commit part. To use this tool, we built a configuration file, ***.travis.yml***, in the root of our repository. And we created a new github branch travis_ci for testing continuous integration.In the configuration, the travis ci will be executed when we ***push*** new code to the remote ***travis_ci*** branch of our repository.

In each execution:
1. A unit test program named ***unit_test.py*** will be executed at first, and the result will be recorded and stored in the ***./report/unittest_log.txt***. For unit test we using the ***unittest***.
2. The branch coverage test will be executed by using the ***Coverage.py***. The coverage test report will be stored in the ***./report/branch_coverage_report*** as html reports.
3. Then the ***travis ci*** machine will push the new ***unittest_log.txt*** file and the new coverage report files to the remote ***master*** branch of our repository with the github access token that has been configured.
4. The static analysis report will generate in the pre-commit part, and the static analysis report will be push to the remote ***master*** branch separately.

4. A link to the github repository where your entire codebase resides. Tag the revisions that were shown in the demo.

GitHub link of our demo (this version is what we showed in the demo):
https://github.com/Debug1995/JoinMe