

Schedule++

Team: Scoreoverflow

Cindy Le(xl2738), Henry Xing(hx2209), David Wan(dw2735), Zichuan Wang(zw2395)

The link to codebase is <https://github.com/Zichuan-Wang/ScoreOverflow/>.

The link to the version shown in the demo is

<https://github.com/Zichuan-Wang/ScoreOverflow/releases/tag/v0.2>.

We demoed on Dec 3. around 11:00 to our IA Siddharth Ramakrishnan during his office hour. When demoing we had some problem with the resolution affecting the UI size. On high resolution displays the UI will be very small due to scaling issues. We will take a look at the problem and explore potential fixes in the final iteration.

This time, we focus on the additional functionalities of the more advanced user groups. The following use cases are shown. First we show the login feature (use case <1> ), which is the same for all users. Use case <2>, <3>, and <5> (search, reserve, and cancel) are the same as the first demo, but during our demo, in order to show use case <4>, the override feature, we first searched a room and reserved as a normal user in order to prepare for override. We then demonstrate use case <4> by going through the entire process as described in this use case, and show that an email was indeed received. We then move on to show use case <5> for the program supervisor with the reservation in batch. However, we did not implement the last two use cases yet, because they are part of the administrator user stories, which we are currently doing manually by hand (writing mysql commands ourselves). We will include these in our final demo.

This time, our CI has the additional functionality of looking for instruction coverage and branching coverage. We used jacoco for this. Jacoco coverage analysis will be run during maven builds, which means the coverage tool is run in both pre-commit CI and post-commit CI. The report produced in post-commit CI will be added back to the repository. As advised by our IA mentor, we will add a coverage percentage goal in the maven build process so that we can make sure everyone writes tests for the new code/features he/she adds. We have explained during our demo why coverage was low: (1) UI testing has many alert messages, which we disabled using conditionals. They make up of a large portion of the missing branch coverage. (2) We have tried, as suggested by our TA, to skip some statements/methods, but Jacoco does not support this functionality. It can only skip classes, which is not what we want.