Retrospective Activity (Team 2)

- We are not meeting as much as we should (improve communication)
- We have been doing work twice as we haven't had a collective understanding of which JUnit we are using, and how objects were passed between activities as examples. A fix could be to have an extremely detailed document for expectations.
- Conflicts are a larger issue; it adds up to become longer the more you wait to pull main into your branch and merge your new changes. A fix could be shorter developer tasks that are done in short bursts of time to minimize conflicts between tasks.
- We had several issues regarding which Android Studio everyone was using and which Gradle version everyone was using. At the end of the iteration, we were able to settle on one version but it has hurt us in most of iteration 2. A fix would be a collectively accepted Android Studio version and Gradle version that we applied already near the end of iteration 2.
- Some of us did not have an understanding of SQL and therefore only some of us were able to work on HSQLDB. A fix could be to have a primary developer and a co-developer on these sorts of tasks to share knowledge and skills.
- Tests were not as good as they had to be due to people focusing on their specific part of the project without understanding the overall idea and functionality of the project.
- Components were largely independent which lead to the confusing implementation of features. A fix could be communication on placements and design choices.

One aspect of your project development that has not been as successful:

Originally we had many different ideas for the project that was larger than we had realized. These user stories were too large and out of scope making us unable to complete them in the expected time. As for iteration 1, we chose the open-auction type of bidding, but later during iteration 2, we switched to sealed auction bidding. These changes helped us simplify the project however they had hampered our progress largely in iteration one and still took time to resolve the issue in iteration 2. A solution that could be used to prevent this from happening again would be deeper discussions on our Discord server between the team on design choices and possibly use the planning poker game with the team to confirm estimates on the user stories. When the estimates are too large, then we could have broken the task into smaller tasks or shrunk the task altogether with a more straightforward task that is similar. The way that we could measure the success would be to take the estimates for the updated user stories and match them up with our team's log and see if the time put into the task matched or was relatively close to the estimate.