

## Criterion E: Evaluation

Meeting criteria for success:

- Online/offline registration: offline registrations work, but the online portion (applet) is not deployed – time constraints and allocations meant that that functionality had to be foregone
- Input current tutor/tutee information (by the coordinator): similar to offline registrations, so this works
- Suggest/make pairings when needed: can filter all tutors for a specific tutee, but no further pairing algorithm is used
- Option for email/print notification (when a pair is made and for attendance alerts): this is not at all met – once again, time constraints and allocations meant that this could not be met
- Records attendance: works, but only with student number
- Sign-in of tutor and tutee at each session (part of attendance): works, and prevents double sign-in
- Manage a database of tutors and tutees: can interact with the database well
- Online and offline portions of the solution can communicate with the database: no online portion

Suggestions for further improvements:

The first thing to do as an improvement would be to add the planned network capabilities of the applet and session program. This would allow for a greater use of the program beyond managing a database, and put the applet into action.

My client suggested that I could have a GUI for the session program. While this would be a very nice quality-of-life change, it does not add much more functionality than is already present, and this improvement would take very long as well.

As I looked through my code, I noticed many inconsistencies and inefficiencies, mostly due to bad design. My design process was very general and mainly focused on the features and functionality of the solution. I should have done some designing for the code itself, with class structures and how I would organize my methods. As an improvement, I would really like to go into my code and reorganize a lot of it to make it more expandable and overall feel cleaner. Also, the way I went about a lot of my algorithms was not very object-oriented, as Java is meant to be. I could improve my solution by working more with the classes and perhaps having a better class structure.

Perhaps my choice to use Java was not the best, considering the difficulty in Java of making any sort of GUI, as I could have done for both parts of my solution. However, the class structure of Java is more suited to an object-oriented solution.