

1. **Add TA Email Addresses** - at the moment, the user interface only has a name field for the teaching assistants. We want to provide a means for entering a textual email address as well. Note that a TA email address should be listed properly in the TA table, but is not needed for the office hours grid. Note that you should employ Java's Regular Expressions library to ensure the email entered by the user is valid. Note that that you may employ code from this [Stack Overflow Discussion](#) on how to do so.
2. **Add a Second Language** - Currently the application only works in English. Add a second language (note, you'll need an additional XML file) such that one can choose between English and the language of your choice. Note that you'll need to provide all labels, prompts, and messages in that second language (just use Google translate if you don't speak another language).
3. **Add Name & Email Text Field Prompts** - text fields these days typically provide ghosted prompts inside that tell the user what they are to enter. Add this as a feature in part through the Desktop Java Framework (i.e. update this framework) such that such prompts can also provide language independence and can be loaded from the language-dependent XML files in a uniform way.
4. **Add Time Slots Column** - add another table to the Teaching Assistant table that will count the number of time slots each TA has been assigned to. A time slot is a 1/2 hour block in our office hours table.
5. **TA Office Hours Toggle** - currently the user may add a TA and then select a TA and then click on the office hours on the right but nothing happens. Change the program such that when you click on a cell in the table on the right it will add (or remove) that TA name to that cell and increment (or decrement) the number of time slots assigned for that TA.
6. **Undo/Redo** - all actions that change the nature of the data, like adding TAs and changing office hours, should be undoable and thus redoable. This means you'll need to employ the **jTPS** library for this.
7. **Saving & Loading** - note that all the changes we've noted should take hold when the user saves/loads files, including TA email addresses. The user will still be saving/loading to/from JSON files. Saved files must use the same file format as [CSE219 TAs.json](#), which you should examine and try to figure out.
8. **Workspace Stylizing** - below is an example of how the application should look once the **CSE219TAs.json** file is loaded. Update the workspace

colors, fonts, and spacing to get your application looking like it. You should try to get exactly the same fonts, color, etc. Note that you'll be using CSS for much of this.

9. **Failure Feedback** - note that when the user tries to add a new TA they must make one that has both a unique name and unique, valid email. Should they fail in either task they must be notified of why what they did was wrong. Provide a message dialog telling them when they provide illegal input.
10. **Fool Proof Design** - One of the best ways to prevent the user from doing illegal operations is to enable/disable controls in order to prevent the user from having illegal options. So, make sure that your application employs this approach. This means disable the Add TA button if there is no text in either the name or email text fields, for example.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder