Scheduler

Old VB Scheduler:

* Checks a file on loading, which lists each RA, the number of weekdays they worked, the number of weekends they worked, and the days of the week they can work.
* As inputs, takes a Start Date and an End Date.
* Creates an array of RAs, with their availability.
* Prioritize1() takes the list of RAs, and a date, and goes through every RA on the list and compares their availability to the day of the week of the date argument; if the RA is available, it returns that RA if they have worked less than the max weekends/weekdays (according to if the date argument is a weekday or weekend), otherwise moves to next RA. If the RA is not available, move to next RA.
* Prioritize2() does exactly the same as Prioritize1(), but it takes an RA as an arguments as well, and only returns an RA if it is not the same as the RA arugument.
* OutputButton\_Click writes to an output text file:
  + It goes through each date between the input Start Date and input End Date.
  + For each date, it finds an available RA using Prioritize1(), and a second available RA using Prioritize 2.
  + It writes the date and the names of the 2 RAs to the file.
  + Then, it goes through the list of RAs, and updates their number of weekdays/weekends (as appropriate) worked where applicable.

Inputs:

1. Individual unavailable dates for RAs (when editing RA)
2. Ranged unavailable dates for RAs (when editing RA)
3. Unavailable days of the week (when editing RA)
4. Ability to edit number of weekends and weekdays worked (when editing RA)
5. Start Date and End Date

Calculations:

1. For each RA, create the list of dates they are available for, between the Start Date and End Date.
2. Iterate through each date between Start Date and End Date; for each date, iterate through RAs, find 2 who are available on that date, while keeping in mind the maximum weekdays and weekends already worked (look at predecessor’s algorithm) (perhaps create an object for each date that contains the date, first RA, second RA, any remaining available RAs, then return a list of such objects)

Output:

1. A calendar view of duties scheduled between each Start Date and End Date, with the 2 top RAs for each shift, followed by the remaining available RAs
2. Totals of weekends and weekdays worked if schedule is accepted (has to be accepted for changes to be implemented)

CHECK SINGLETON REQUIREMENT

Database values:

John - can't work weekends, and July 8 and July 23

Sam - can't work weekdays, and July 3 and July 18

Maggi - can't work weekdays, and July 4 and July 17

Maddie - can work everyday

Graeme - can't work weekends, and July 9 and July 22

Yash - can't work on any day, and July 4 and July 5

DATABASE IMPLEMENTATION:

Need 3 tables:

1) Lists information on RAs

2) Connected table (by ID) that stores unavailable days of the week

3) Connected table (by ID) that stores unavailable dates

For each Swing component, ui.gen decides only the visual aspects, such as size and name, not data. That is accessed, dealt and manipulated by the ui half.

Ideas:

Advanced settings that allow:

1. Deselection of particular RAs
2. Ignorance of any previous data on weekends and weekdays worked, and simply attempt to create an equal schedule within the start date and end date

Identify when adding/editing RAs, what the error is, according to the user’s input, and inform the user accordingly.

Add labels to everything.

For all lists, see if there is a way to update the tables without creating a new table and adding it each time.

Remove unused variables and imports

Try to make WorkPanel handling of updating lists after selection better, try to refrain from passing so many arguments to listener.

Handle exception using dialog message boxes, not console error messages.

Have a refresh button next to all the table views to reobtain data from database

Deleting RAs automatically deletes their unavailable dates and days upon Refresh

Edit RA to edit unavailable days and dates for them

Stage gives a preview of the results and changes, allowing cancellation/rollback.

Commit implements them

Creating an RA puts them into the database with an incremented ID.

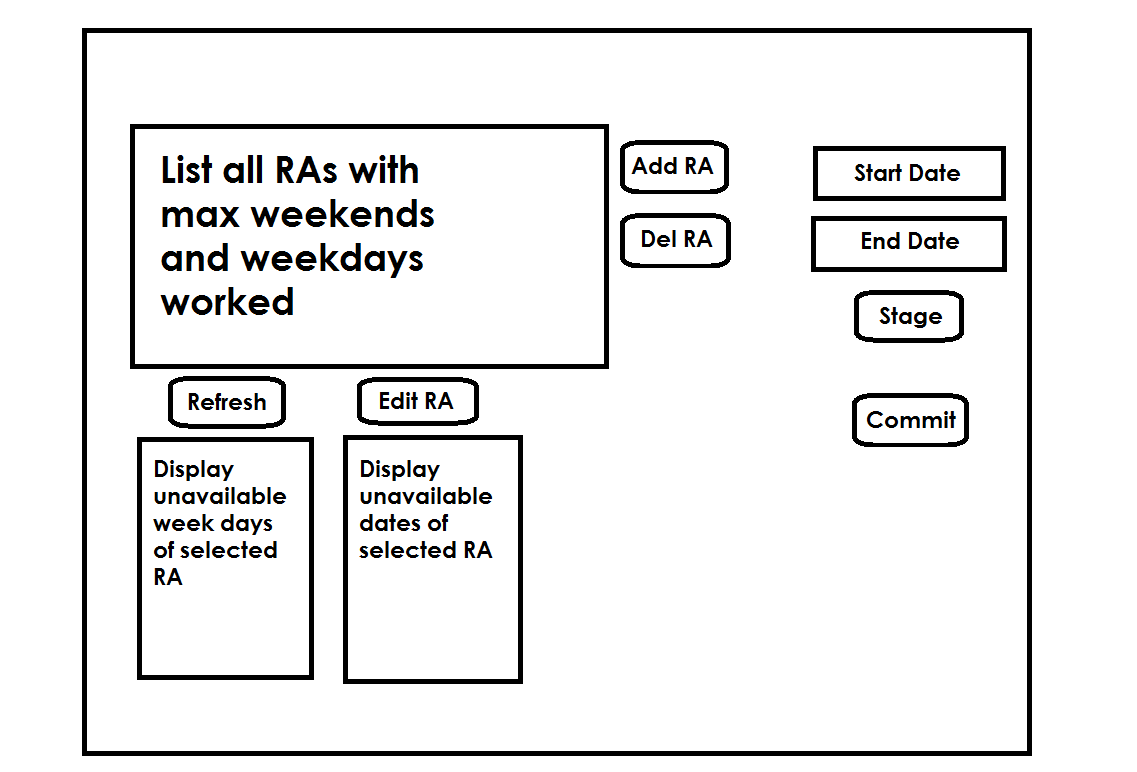
Check if run works without refreshing (should).

Lookup adding runtime exceptions for actionPerformed when dealing with SQL code.

Fix all tableName problems

Maybe change displays to show caps for unavailable days.

Perform all checks for buttons, especially where they need data (such as selected row or a required name field) to work.



Refactor method arguments and parameters (example table names) to allow maintenance.

Fix bracket indentation

Add all used bookmarks/web resources to documentation.

C:\Users\Yash\Eclipse\_workspace\_150521\Scheduler