Criterion B: Design

The solution will incorporate three databases:

- One for the tutors registered for the program
- One for the tutees registered for the program
- One for the matches made (for the “active” tutors/tutees)

<u>For the offline portion that will take care of managing of the database and attendance:</u>
The offline portion will have 4 classes:

- Student
- Tutee
- Tutor
- Match
- TutoringDBMS (database management system)

Student is the superclass for classes Tutee and Tutor. Student will have instance fields that hold all of a student’s properties, including name, student number, gender, days free, etc.

Tutee and Tutor and the subclasses of Student. The only differentiation are the classes themselves. Classes Tutee and Tutor have no additional instance fields to those of class Student. Classes Tutee and Tutor have static methods which allows the main method of the solution to interact with the files containing data for all tutees and tutors in the program.

Match is the class that links classes Tutee and Tutor, with instance fields like the tutor in the match, the tutee, the meet days, and the courses being tutored. It has static methods which allow the main method of the solution to interact with the text file containing the data for all the matches that are currently active.

Class TutoringDBMS is the class that puts everything together – it has the main method of the solution, as well as supporting static methods which allow matches, tutors, and tutees to be created and deleted. The operation of the main method is planned to follow the flowchart in Fig.1. It serves as a tool for tutors/tutees as well as the peer tutoring coordinators, depending on where and when the “password” is entered. The coordinator menu is the menu presented to the coordinators themselves which allow for the access and changing of the databases. The operation of this coordinator menu is outlined in Fig.2.

<u>For the applet portion</u>
The applet will be used solely for registration purposes for prospective tutors/tutees. It will be created using java’s swing API, which allows for use of a number of GUI components. The applet will collect all the relevant information for the registration of a tutor/tutee, and will validate the form (no invalid information/input) before it can be submitted.

Fig.1: Flowchart for operation of main method in class TutoringDBMS:

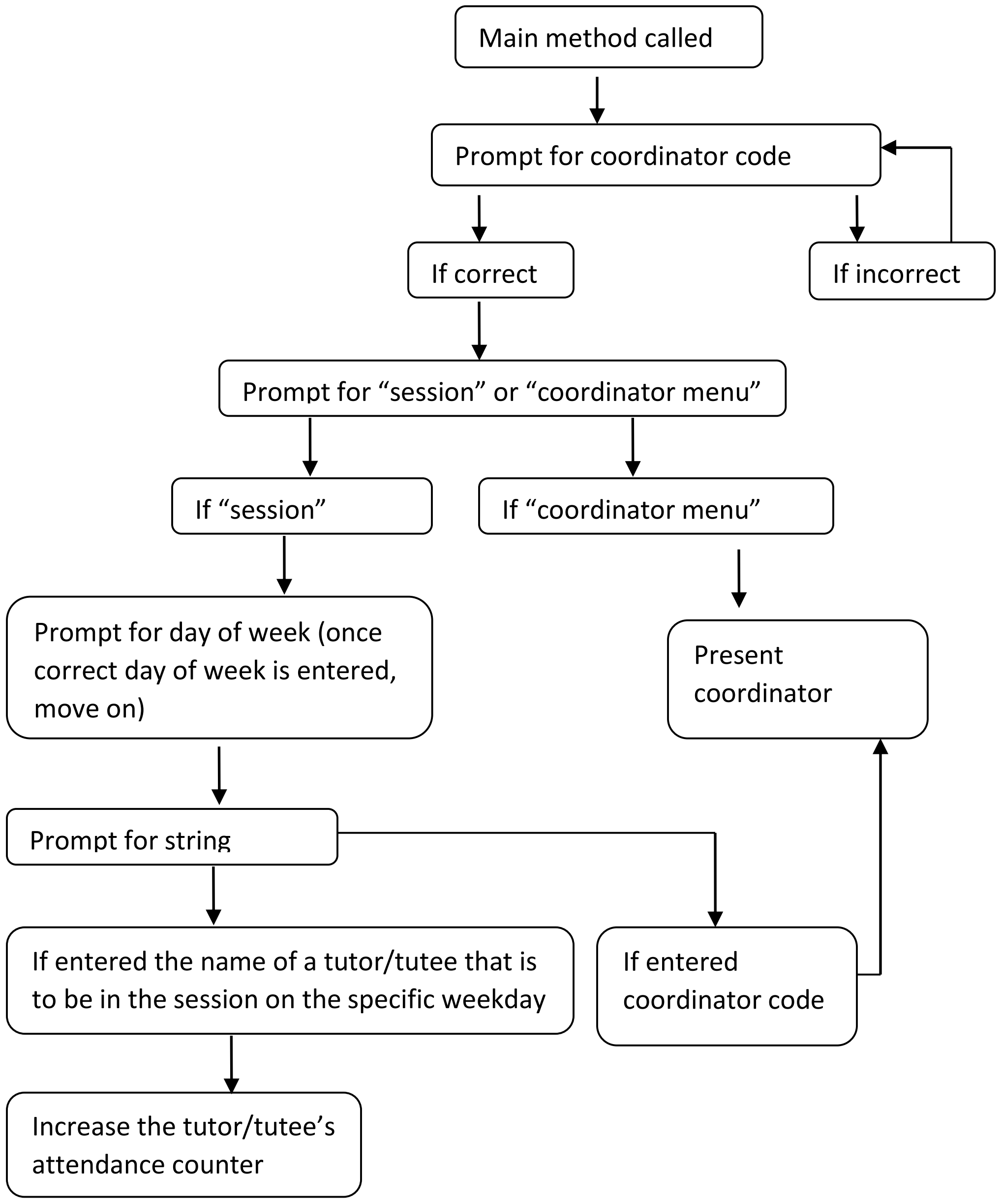
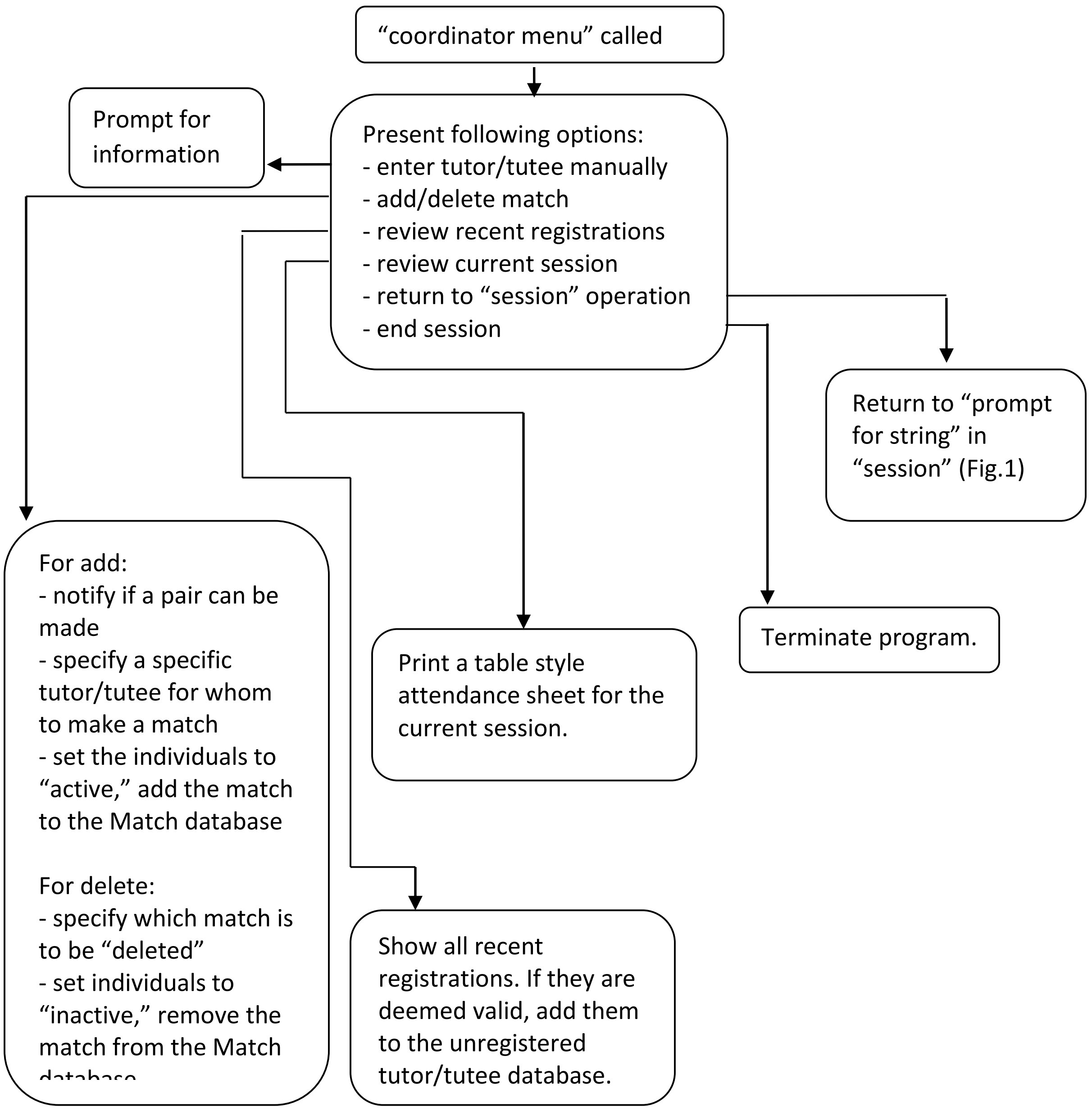

Main method called
Prompt for coordinator code
If correct
If incorrect
Prompt for “session” or “coordinator menu”
If “session”
If “coordinator menu”
Prompt for day of week (once correct day of week is entered, move on)
Present coordinator
Prompt for string
If entered the name of a tutor/tutee that is to be in the session on the specific weekday
If entered coordinator code
Increase the tutor/tutee’s attendance counter

Fig.2: Flowchart for coordinator menu:


"coordinator menu" called
Present following options:
- enter tutor/tutee manually
- add/delete match
- review recent registrations
- review current session
- return to "session" operation
- end session
Prompt for information
Return to "prompt for string" in "session" (Fig.1)
Terminate program.
Print a table style attendance sheet for the current session.
Show all recent registrations. If they are deemed valid, add them to the unregistered tutor/tutee database.
For add:
- notify if a pair can be made
- specify a specific tutor/tutee for whom to make a match
- set the individuals to "active," add the match to the Match database
For delete:
- specify which match is to be "deleted"
- set individuals to "inactive," remove the match from the Match

To test the offline portion (the database manager):
The 3 database files (for tutee, tutor, match) will have data put into them for testing purposes.

Action to test	Method of testing
Checks for password	Keep prompting for the password until the password is entered
Allows a day to be selected, and read matches for that day	Select a day and go to coordinator menu to review session. See if the attendance list matches the matches that should be there on the day (check the database file).
Allows sign in of tutors/tutees	Type in student number during the session. Check in database if the attendance number went up.
Disallow multiple sign ins of the same person	Sign in once, then sign in again in the same session using the same student number. Check if attendance number goes up the second time.
Allow transition from "session" to coordinator menu	Go to session. Enter password.
Eliminate day prompt if it has already been entered	Select a day (Monday, Tuesday, etc.). Go to coordinator menu. Go back to "session."
Review of current session including attendance status	Sign in some people for the day, not others. Go to coordinator menu and select review session.
Add new tutor/tutee	Add a new tutor/tutee, then check in "view all tutors/tutees" or in database to see if the new tutor/tutee has been added.
Delete a tutor/tutee	Delete a tutor/tutee, then check in "view all tutors/tutees" or in database to see if the tutor/tutee has been removed.
Ability to display all tutors/tutees	Make the selection to display all, and compare to database
Add a match	Show available tutees and compatible tutors, check if it matches databases. Then when a match is created, check if the days free of the tutor/tutee have been modified, and if the match has been added to the database.
Delete a match	Show all matches and when a match is deleted, check that it has been removed from the database and that the tutor/tutee's days free are modified accordingly.
Proper saving of all data	Whenever a change is made, check the appropriate database to see if data is saved correctly.

To test the applet:

Action to test	Method of testing
Text fields are not empty	Leave some text fields empty. Do a few times with different text fields.
Gender is selected	Keep both gender radio buttons unselected.

Student number is a 9 digit number	Make student number text or an invalid number.
Email contains proper format	Put in emails with no user name, no @ symbol, improper domain name.
Grade is selected	Leave all grade radio buttons unselected.
Tutor/tutee designation is selected	Leave both unselected.
At least one day free in the week	Leave all days unchecked.
No repeated courses	Make some courses the same.
Course combo boxes keep their state when other ones are added/removed	Play around with the combo boxes and adding/removing courses.