

Automatic TA Assignment Documentation

Database Tables and commands to access

- ssh cssl@ibss.usc.edu
- Install XAMPP and start mysql server (Default port : 3306)
- For above, follow <http://hammadk.com/how-to-start-xampp-in-terminal-mac-osx/>
- cd /Applications/XAMPP/bin
- ./mysql -u root -p
- Enter Password
- use ta_project;
- select * from User; // table to store the User info i.e; login details
- select * from TA; // table to store the TA background info linked with a User
- select * from Course; // table to store the courses info
- select * from Course_Section; // table to store the course section info for each course
- select * from TA_Preferences; // table to store the TA preferences for each section
- select * from TA_Time_Constraints; // table to store TA time constraints
- select * from Reason; // table to store reasons for non-availability at a time.
- select * from Milestones; //table to store milestones and their ranking
- select * from Time_Intevals; //All the unique time slots available in the system
- select * from Admin_Matching; //Table to store admin matching rules
- Select * from Matching; //Table to store Matching performed by algorithm.

PHP and Java Jars (Apache server)

- cd /Library/WebServer/Documents/php/
- Java 1.8 should be available;
- The php folder should be in the apache server default folder.

Web Portal

- Url: ibss.usc.edu : leads to login page
- New Users click signup and signup with name, username and password
- Others can login to enter User Dashboard
- User Dashboard: Three tabs
 - Background
 - Add Personal Details: add details related to classes taught before, milestones reached, area of study, number of semesters of teaching assistant experience etc
 - View Personal Details: view details
 - Update Personal Details: update details
 - Add Time Constraints: Choose all time constraints in the sections and add more if needed. (Unless all time slots are added, overlaps may exist. For

eg. Choosing MW 10-2 as a constraint is different than choosing M 10-2 and both options need to be chosen to avoid any overlaps)

- View Time Constraints: Can Update and Delete any previous time constraints
- Courses
 - View Courses: view all active courses and sections for each course
 - Submit Course Preference: submit preference for each section(courses with lecture codes for options) between the ranges High to Low. Need to submit preference for all courses.
 - View Submitted Preferences: View submitted interest level for each course-section. Can Update Preference for a course section- change interest level between ranges High to Low.
- Matching
 - View Matching: Lists the course code- lecture code- lab codes, timings of class for sections allocated to the user. Can be viewed only after admin releases matching.
- Admin Dashboard: Three Tabs
 - Users
 - View Users: Can see users, their ta ids, their areas. The action button allows admin to view user details (like preferences), activate or deactivate the user, update a user's details and delete the user
 - View User Preferences: view users and their preferences for each course section along with their interest level. Admin can delete a user's preference.
 - Courses
 - Add courses: fill out course name, course code, how many half and full tas needed
 - Add sections: choose a course, and add either a lecture or a lab and include it's timings as MW,TTH, MWF or M,T,W,Th,F. Also add lecture and lab codes. Lab codes can be null (all others are required fields)
 - View Course Details: can see each course, area, active or inactive. Action button allows to view, update, delete, activate or deactivate a course. By clicking view for a course, admin can also see all active sections for that course
 - View Section Details: View a list of all sections with timings, course codes, lecture and lab codes
 - Matching
 - Add TA-Course (Admin Override): allows admin to assign an active ta to an active section
 - Perform matching:
 - Allows admin to deactivate or activate any previous matching (admin or algorithm generated)

- Run matching runs the matchings by the algorithm. Button click runs the java file and populates matching table
 - View Matching: shows tas and the sections they are assigned to. Also shows source of matching(admin/algorithm) Admin can activate / deactivate or delete a matching
 - Release matching: releases matching for users to see
 - Block matching: blocks matching results to prevent users from seeing
 - View matching stats: shows the count and source of users matched, sections matched. Names the users that are yet to be matched and the sections that have no TA assigned
- Users once signed up need to add personal details first before they can add course preferences or time constraints (this check has been implemented)
- All code is in the server at location: /Library/WebServer/Documents/
- Folders: php, java, html, css, js, db, images

Algorithm Outline:

- Populate maps with id and objects from each of the tables;
- Get the admin matching and previous matching that is active.
- Remove TA and course sections of above matching from available pool.
- Calculate score of each ta based on milestone, no. of years of ta experience.
- Populate the score in preference object for the TA : ta_score + if_happy_and taught this course_before(0.2) + if_its_Quant_course,Quant_Student add 5(very high score)
- For all the preference list in decreasing order of interest,
 - Eliminate preference object where TA time and Lecture Time and/or Lab Section Time clash;
 - For each section, sort the list of eligible TA's based on score.
 - Modify this list by bringing forward the TA who taught this course and is happy and move back the one who is not happy and taught it last sem.
 - After this we get the list of eligible TA's for each section in decreasing order of score.
 - Distribute these TA's one by one to each of the section.
 - Keep removing the allocated TA and section from eligible list.
- This will give us one result of matching, if unsatisfied with any, remove them from UI and run the algorithm again.

Db link from java code : config.xml

Driver class : GenerateAssignment.java