Automatic TA Assignment

Rajdeep Kaur
Srinidhi Nandakumar

Introduction

- The application assigns TAs to sections based on the experience, background and preferences of the TAs.

- It consists of two main parts.

- The front end allows the User and Admin to access the Web Portal to perform various functions

- The backend uses the information fed by users and admin to match teaching assistants signed up with the portal to courses added by the admin

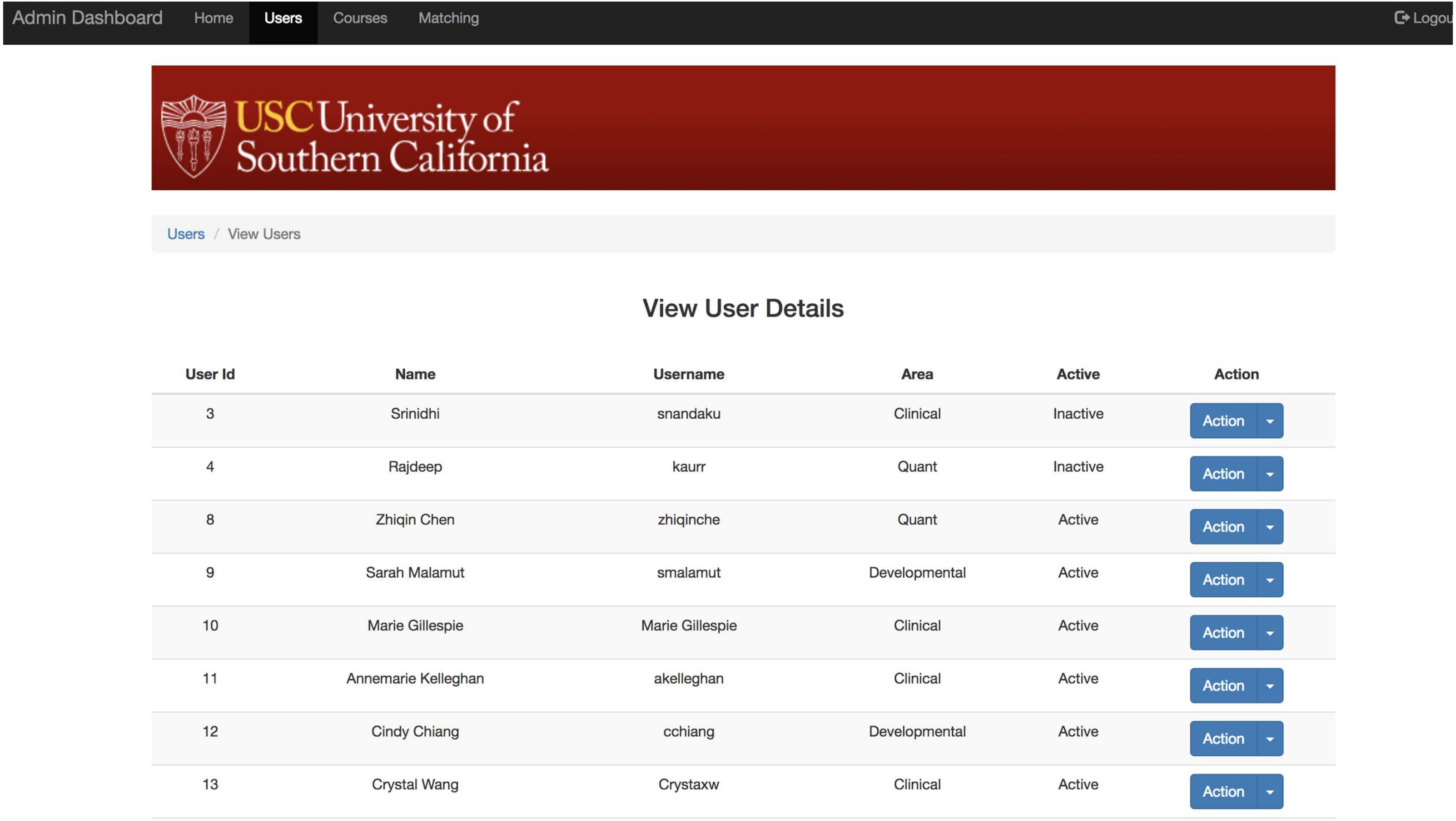
Web Portal

Admin Dashboard

- User
  - View Users
  - View User Preferences
- Courses
  - Add Courses
  - Add Sections
  - View Courses
  - View Sections
- Matching
  - Admin Override
  - Perform Matching
  - View Matching
  - Release/Block Matching
  - View Stats

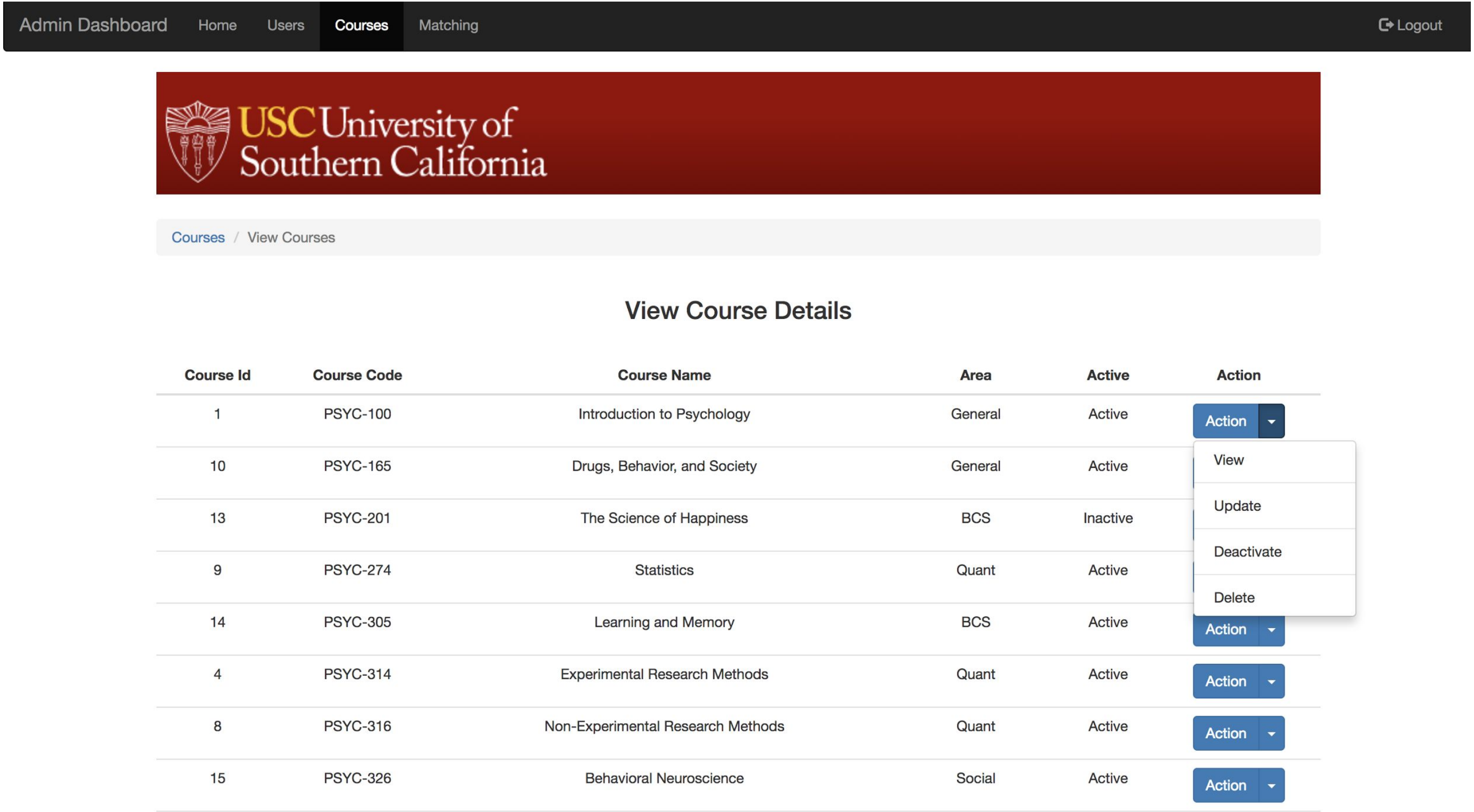
User Dashboard

- Background
  - Add personal details
  - View Personal Details
  - Add time constraints
  - View time constraints
- Courses
  - View Courses and Sections
  - Submit Preferences
  - View Submitted Preferences
- Matching
  - View Matching



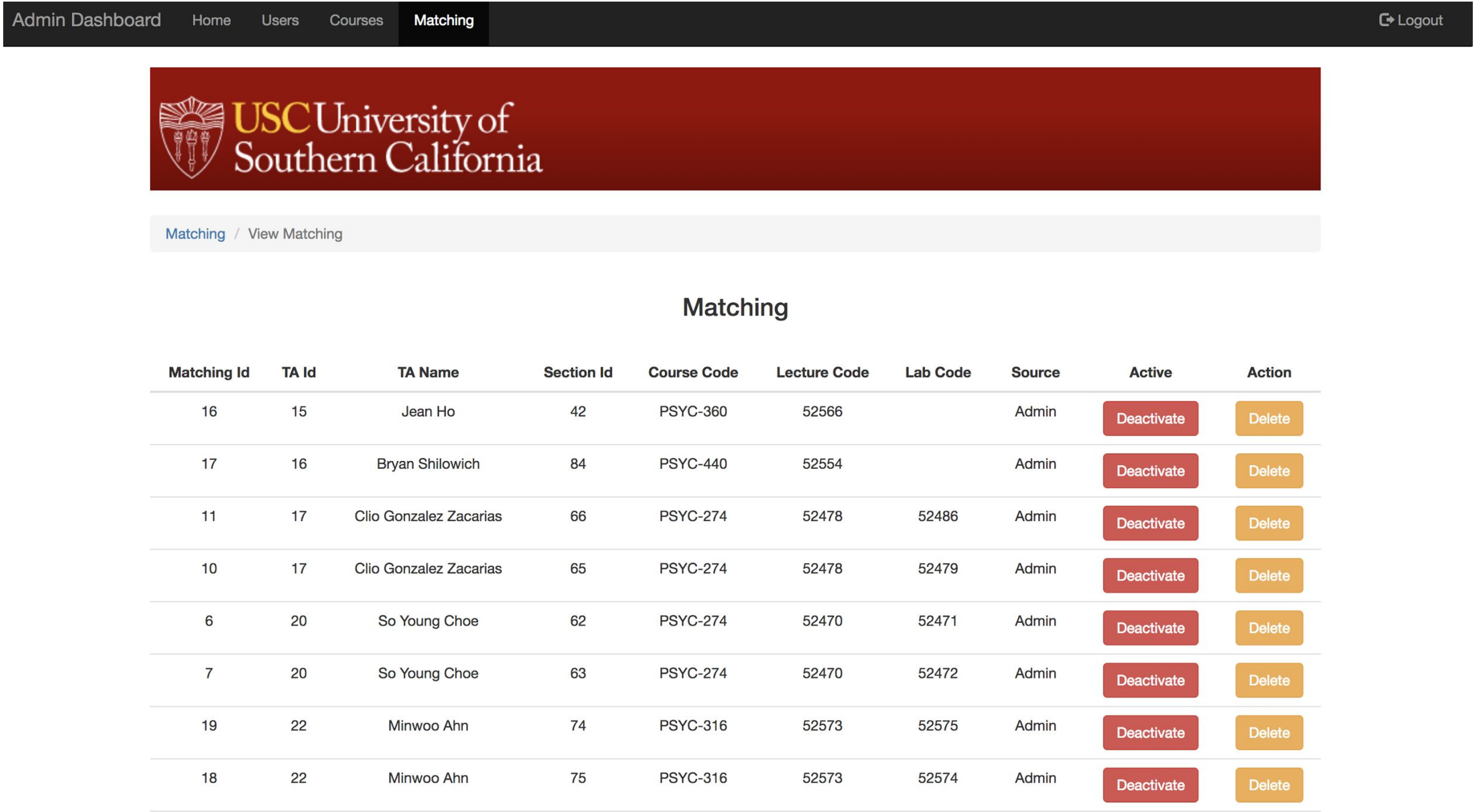
View User Details

User Id	Name	Username	Area	Active	Action
3	Srinidhi	snandaku	Clinical	Inactive	Action
4	Rajdeep	kaurr	Quant	Inactive	Action
8	Zhiqin Chen	zhiqinche	Quant	Active	Action
9	Sarah Malamut	smalamut	Developmental	Active	Action
10	Marie Gillespie	Marie Gillespie	Clinical	Active	Action
11	Annemarie Kelleghan	akelleghan	Clinical	Active	Action
12	Cindy Chiang	cchiang	Developmental	Active	Action
13	Crystal Wang	Crystaxw	Clinical	Active	Action



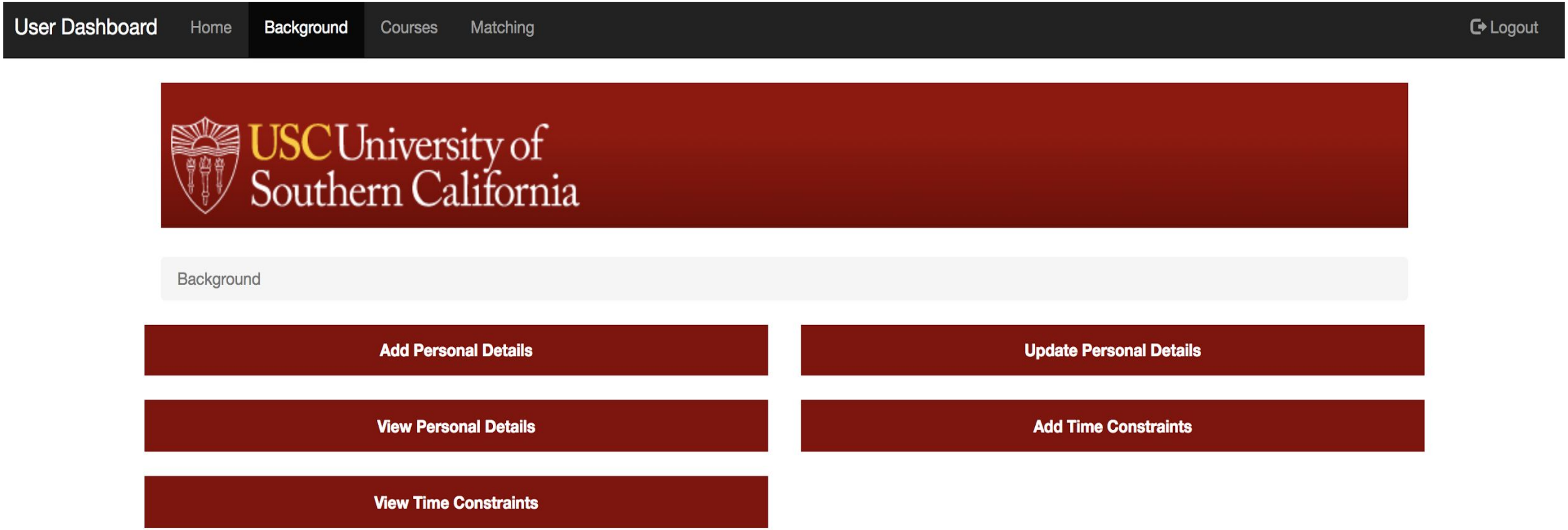
View Course Details

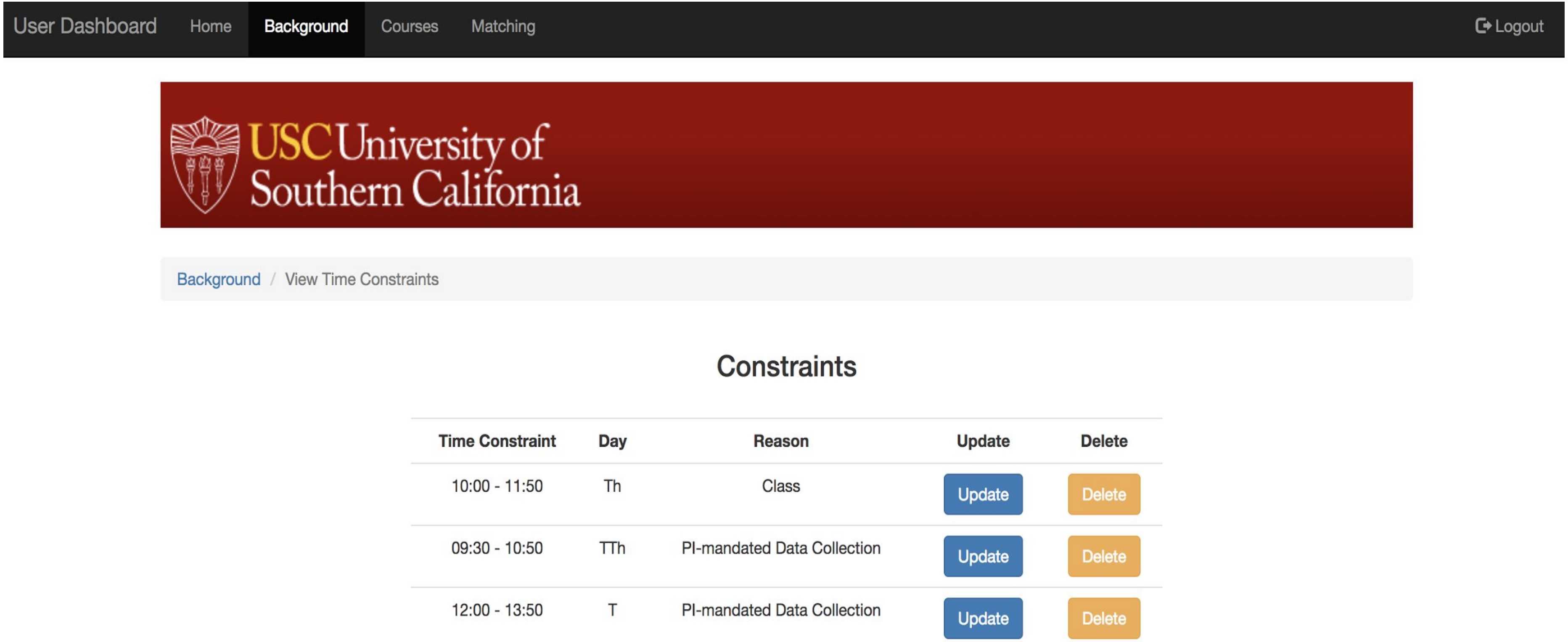
Course Id	Course Code	Course Name	Area	Active	Action
1	PSYC-100	Introduction to Psychology	General	Active	Action
10	PSYC-165	Drugs, Behavior, and Society	General	Active	
13	PSYC-201	The Science of Happiness	BCS	Inactive	
9	PSYC-274	Statistics	Quant	Active	
14	PSYC-305	Learning and Memory	BCS	Active	Action
4	PSYC-314	Experimental Research Methods	Quant	Active	Action
8	PSYC-316	Non-Experimental Research Methods	Quant	Active	Action
15	PSYC-326	Behavioral Neuroscience	Social	Active	Action



Matching

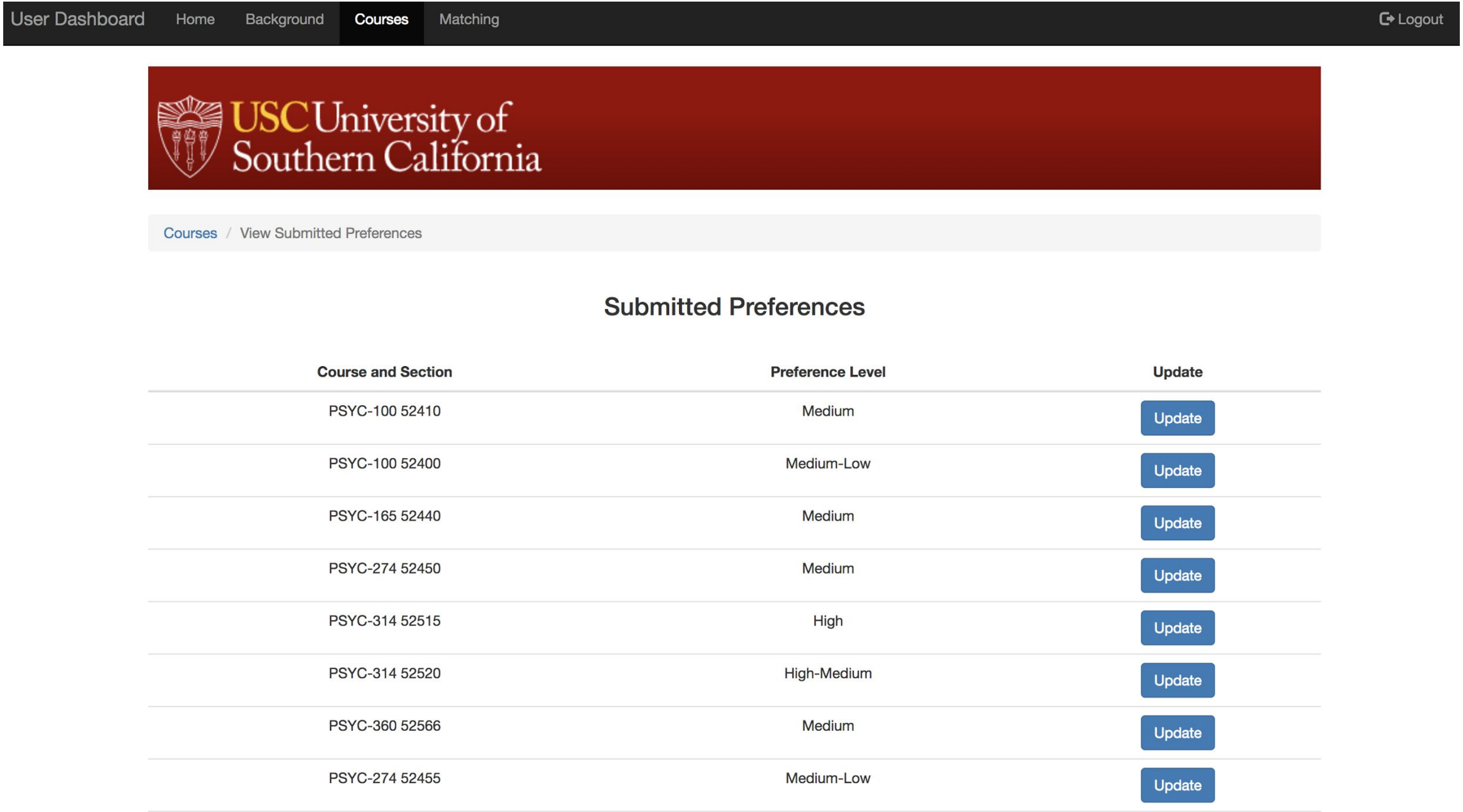
Matching Id	TA Id	TA Name	Section Id	Course Code	Lecture Code	Lab Code	Source	Active	Action
16	15	Jean Ho	42	PSYC-360	52566		Admin	Deactivate	Delete
17	16	Bryan Shilowich	84	PSYC-440	52554		Admin	Deactivate	Delete
11	17	Clio Gonzalez Zacarias	66	PSYC-274	52478	52486	Admin	Deactivate	Delete
10	17	Clio Gonzalez Zacarias	65	PSYC-274	52478	52479	Admin	Deactivate	Delete
6	20	So Young Choe	62	PSYC-274	52470	52471	Admin	Deactivate	Delete
7	20	So Young Choe	63	PSYC-274	52470	52472	Admin	Deactivate	Delete
19	22	Minwoo Ahn	74	PSYC-316	52573	52575	Admin	Deactivate	Delete
18	22	Minwoo Ahn	75	PSYC-316	52573	52574	Admin	Deactivate	Delete


User Dashboard
Home
Background
Courses
Matching
Logout
USC University of Southern California
Background
Add Personal Details
Update Personal Details
View Personal Details
Add Time Constraints
View Time Constraints


User Dashboard
Home
Background
Courses
Matching
Logout
USC University of Southern California
Background / View Time Constraints

Constraints

Time Constraint	Day	Reason	Update	Delete
10:00 - 11:50	Th	Class	Update	Delete
09:30 - 10:50	TTh	PI-mandated Data Collection	Update	Delete
12:00 - 13:50	T	PI-mandated Data Collection	Update	Delete



Courses / View Submitted Preferences

Submitted Preferences

Course and Section	Preference Level	Update
PSYC-100 52410	Medium	Update
PSYC-100 52400	Medium-Low	Update
PSYC-165 52440	Medium	Update
PSYC-274 52450	Medium	Update
PSYC-314 52515	High	Update
PSYC-314 52520	High-Medium	Update
PSYC-360 52566	Medium	Update
PSYC-274 52455	Medium-Low	Update

Back End

DATA POPULATION

- Domain Object in java for each table in database
    - Example: Course.java, TA.java
- Populate domain objects using mybatis and my-sqlconnector.
- Create Map put of lists for each retrieval based on Id.
- Enrich data;
    - Example : Fill TA domain object with score based on milestones.

Algorithm Flow

- 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)

Algorithm Flow

- 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.

Algorithm Flow

- This will give us one result of matching, if unsatisfied with any, remove them from UI and run the algorithm again.
- Or, delete few and add admin override manually.

THANK YOU !!!