

# DATABASE TABLES SCHEMA

## NOTES:

- The naming convention is camelCase.
- All purple variables are countable (substitutes for arrays); front-end should expect functions to retrieve count and actual list.
- Blue boxes are notes for the front-end team; red for the back-end team.
- The first field is the unique identifier for that item.
- Dotted arrows are the same variable for one instance of the item.

minimum 1,  
maximum 2,  
enforced in front-end

generated by front-end  
operations before  
sending to database

## APPLICATIONS TABLE

<b>username</b> (String) (unique)	<b>firstName</b> (String)	<b>lastName</b> (String)	<b>year</b> (String) (1 of 4 options)	<b>gpa</b> (float)	<b>grade320</b> (char)	<b>reference</b> (String) ( <i>'firstname lastname, email'</i> )	<b>attachment</b> (String) (S3 link)
---	------------------------------	-----------------------------	---	-----------------------	---------------------------	--	--

## STUDENTS TABLE

<b>username</b> (String) (unique)	<b>conflict</b> (int = section ID)	<b>ranking</b> (int = 1, 2, ...)	<b>sectionRanked</b> (int = section ID)	<b>matched</b> (int = section ID)
---	---------------------------------------	-------------------------------------	--	--------------------------------------

So, if a student prefers section 7  
and then section 8, the pairings  
would be (1, 7), (2, 8).

Send out notification when this  
value changes from -1 to  
section ID, since the student  
is now matched.

## PROFESSORS TABLE

<b>username</b> (String) (unique)	<b>section</b> (int = section ID)	<b>ranking</b> (int = 1, 2, ...)	<b>appRanked</b> (String = student's username)
---	--------------------------------------	-------------------------------------	--

## SECTIONS TABLE

<b>id</b> (integer) (unique, taken from schedule)	<b>day</b> (String)	<b>time</b> (String)	<b>profName</b> (String) (input manually)	<b>capacity</b> (int)	<b>numEnrolled</b> (int)
--	------------------------	-------------------------	---	--------------------------	-----------------------------

Retrieve capacity and  
numEnrolled from here.

Update numEnrolled  
as the matching  
algorithm progresses.

## MATCHINGS TABLE

<b>section</b> (int = section ID)	<b>username</b> (String = student's username)
--------------------------------------	--

## USERS TABLE

<b>username</b> (String) (unique)	<b>password</b> (String)	<b>type</b> (Boolean; 0 = student, 1 = professor)	<b>firstName</b> (String)	<b>lastName</b> (String)
---	-----------------------------	--	------------------------------	-----------------------------

Name will be asked for  
during registration. This  
name can be retrieved  
from here to greet user.