

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.
- **Yellow boxes** are character limits / hard-coded formats / other restrictions that both teams should be aware of.
- The first field is the unique identifier for that item.
- Dotted arrows are the same variable for one instance of the item.

varchar(50)	varchar(50)	varchar(50)	varchar(10)			varchar(200)	varchar(100)
username (String) (unique)	firstName (String)	lastName (String)	year (String) (1 of 4 options)	gpa (float)	grade320 (char)	reference (String)	attachment (String) (S3 link)
↑			"Freshman" etc.	"firstName lastName, email"			

varchar(50)	conflict	ranking	sectionRanked	matched
username (String) (unique)	(int = section ID)	(int = 1, 2, ...)	(int = section ID)	(int = section ID)

varchar(50)			varchar(50)
username	section	ranking	appRanked
(String) (unique)	(int = section ID)	(int = 1, 2, ...)	(String = student's username)

The diagram illustrates the 'sections' table schema and its usage. The table has the following columns:

- id** (integer, unique, taken from schedule)
- day** (String, e.g., 'MW', 'MWF', 'TuTh', etc.)
- time** (String, e.g., '2:30 PM - 3:45 PM')
- username** (String = Professor's username)
- applicant** (String = student's username)
- capacity** (int)
- numEnrolled** (int)

Usage notes for the columns:

- username**: Used to be String = Professor's name. Simplifies back-end operations. Front-end will have function to retrieve instructor's name when displaying section info.
- applicant**: Every student who has submitted application choosing this section. Update with every submission.
- capacity** and **numEnrolled**: Retrieve capacity and numEnrolled from here.
- numEnrolled**: Update numEnrolled as the matching algorithm progresses.

section (int = section ID)	username (String = student's username)
--------------------------------------	--

varchar(50)	varchar(25)	type	varchar(50)	varchar(50)
username	password		firstName	lastName
(String)	(String)	(Boolean; 0 =	(String)	(String)
(unique)		student, 1 =		
		professor)		

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

minimum 1,
maximum 2,
enforced in front-end

generated by front-end
operations before
sending to database

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.

Retrieve capacity and numEnrolled from here.

Update numEnrolled
as the matching
algorithm progresses.