

Project team 1 - TARA - Quintessential TA & RA finder

Team Members:

Prasanna Lalingkar, Rajdeep Rao, Sachin Badgujar, Chetan Borse

Taiga Project Name:

SSDI-FALL-2016-TEAM-1

PROJECT PROPOSAL

Overview and Motivation

Currently there is no Standard, Uniform and Equal Opportunity system in place at UNC Charlotte to find Teaching/Research Assistants. We believe that it is imperative to find the best candidate for a position by bringing the demand and supply together. In the process we also aim to introduce more transparency in the system and encourage greater collaboration between the Professors, University and Students.

Scope and Objectives

We intend to create a platform to be used by the Students to find opportunities, by the Professors to scout talent and the university to manage resources.

Student

Each Student will create a profile. The student profile will have his entire information (personal, professional, academic, skills, opportunities he is looking for etc). The student can browse TA and RA positions postings (open and archived both). The student can apply for open positions provided he clears the prerequisite conditions set by the professor.

Professor

Each Professor will create a profile. The professor can list out his interests and academic/research information on his pages. The Professor creates an opening for a position in his group. The Professor can set a test (MCQ's, Descriptive questions, Behavioural questions, Case Studies, Programming tasks etc) and allow students to work on the test. The Professor can then offer the position to a student who meets his criteria.

University

The University/Department can keep track of the positions and fulfill necessary administrative tasks associated with this hiring.

Business Logic

We intend to create a smart system which will rank students who apply for a position based on

1. Their past experience and skill set.
2. Their performance in the tests set by the Professor.
3. Recommendations and Ratings provided by Professors with whom the student has worked previously.

The professor can then offer the position to the students who make it to the top of the list. We intend create a rating system (for both the Students and the Professors) which will be factored in during all the decisions made.

The students can benefit from this transparent system by getting what they deserve and the Professors can pick the best among the students to get their work done.

PROJECT ENVIRONMENT AND ARCHITECTURE

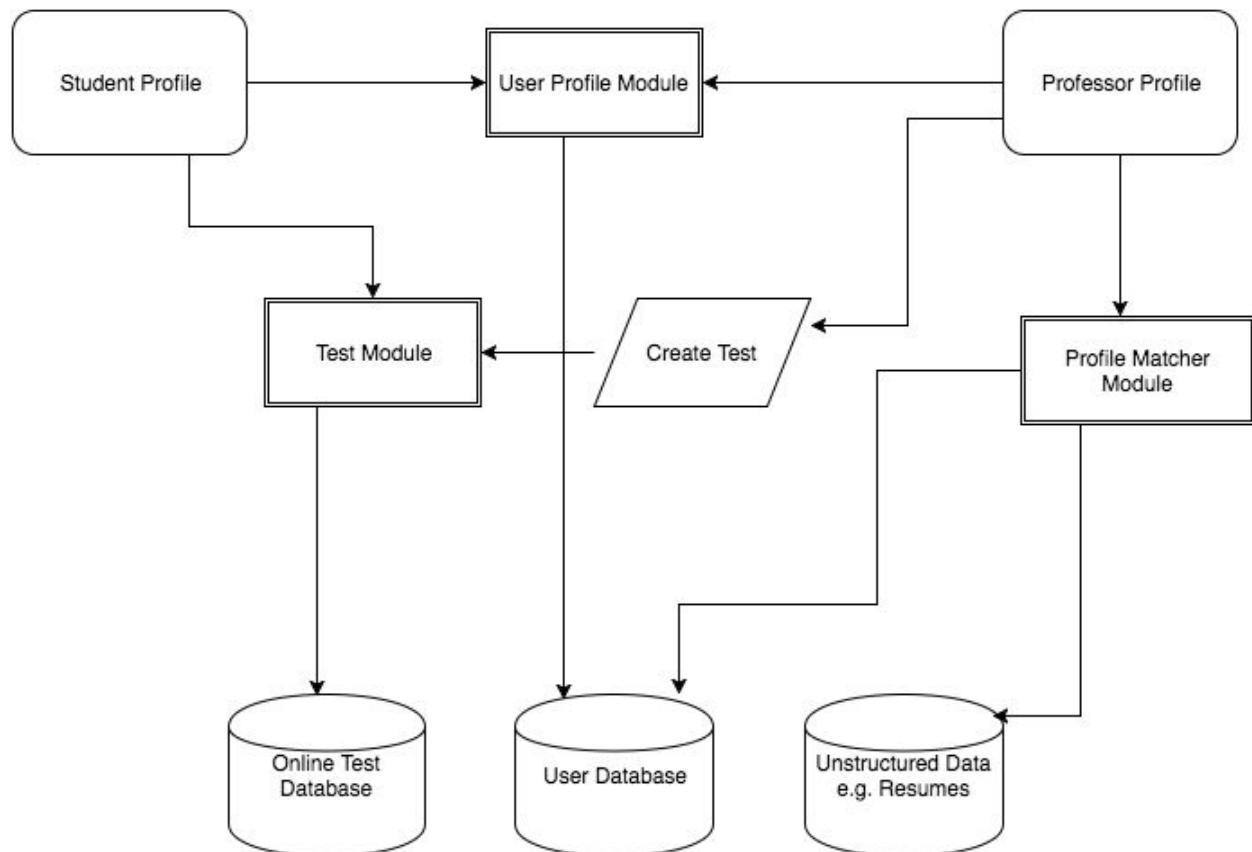
Environment

Project TaRa, for the most part is going to be implemented in Java, with the following technologies:

1. Client-side:
 - **HTML:** To create views, HTML templates would be required that would allow for the content to be displayed on the web browser. We would be using the Atom text editor for web development
 - **CSS:** To make the content visually appealing, CSS would be embedded into the HTML code
 - **Angular JS:** Angular JS is essentially an MVC, that provides us with numerous advantages over other conventional web development frameworks by allowing two way data binding, modular development amongst other amazing functionalities and it's also a great way to create single page applications. Given How TaRa would be heavy on the front end, these are a few reasons that convinced us that Angular JS would make for an appropriate front-end framework for this project
2. Back End web development:
 - **Spring boot** has been chosen for the backend, mainly because we only need to model table structures as classes and provide configuration parameters to connect to databases as opposed to the conventional arduous process. Spring tool Suite will be used for the backend web development.
 - **SQL** is the database of choice, since each user profile that's created is structured uniformly that allows us to exploit the simplicity of sql databases.

3. Web Server:
 - **Apache Tomcat** Since this project is so Java Centric, an Apache tomcat server was set up without much of a hassle. The web server would be deployed via the terminal.
4. Testing Framework:
 - **JUnit** is the unit testing framework that has been chosen, again since the project is developed mainly in java, Junit seems like the best choice in the market. Again, STS would be used for this, since it can be extended to the existing STS IDE so easily.
5. Version Control source repository:
 - **Git** is the most appropriate version control system, since we're all working on different operating systems and on different modules that a seamless interaction is made possible through github. Also, it allows for maintaining logs that allows for monitoring each of the group member's contribution.
6. Project Management tool:
 - **Taiga** is an open source project management tool that allows us follow agile practises and tracks progress of the project.

Architecture



PROJECT REQUIREMENTS

User Stories

ID	Subject	Description	Sprint	Total Points
1	As a Student, I should be able to create a profile on TaRa so that I can access the Web Application.	The student profile will have his entire information (personal, professional, academic, skills, opportunities he is looking for etc).	Sprint 1	5
2	As a Professor, I should be able to create a profile on TaRa so that I can access the Web Application.	The professor can list out his interests and academic/research information in his profile.	Sprint 1	5
3	As a Professor, I should be able to create a online test so that students who have been shortlisted can attempt it.	The Professor can create a online test (MCQs, Descriptive questions, Behavioural questions, Case Studies, Programming tasks etc).	Sprint 2	12
4	As a student, I should be able to view open and archived TA/RA positions so that I can apply to positions of my interest	A student will log-in and can browse all the TA/RA positions based on the filter criteria.	Sprint 2	6
5	As a Student I should be able to apply to open TA/RA positions so that my profile can be evaluated by the Professor		Sprint 3	6
6	As a Professor, I should be able to manage the positions created by me, so that I can take necessary action.		Sprint 3	4
7	As a Professor, I should be able to shortlist students for a position so that I can set a test for them.		Sprint 1	4

Tasks

ID	Subject	Description	User Story	Sprint
8	Create the front end for login page	The login page should have a Email ID (uncc email id) and password field along with submit button	1	Sprint 1
9	Create the front end for signup page	Create a signup page which should include a text box for email id (uncc), password and confirm password each. On successful signup the user should be logged in and taken to his home page	2	Sprint 1
10	Create the front end for student profile - home page for a student after successful login		7	Sprint 1
11	Create the front end page for a professor profile - home page for a professor after successful login		7	Sprint 1
12	Create the databases for People, Student, Professor	The tables would have fields as discussed in design and appropriate primary and foreign keys to link them together	1	Sprint 1
13	Write resumer matcher module	Logic to match the student skills to the requirements of a position	7	Sprint 1
14	Design front end for viewing shortlisted students.		7	Sprint 1
15	Design database to store TA/RA position requirements.		7	Sprint 1
16	Perform acceptance tests for creating new profile for Student.		1	Sprint 1
17	Perform acceptance tests for creating		2	Sprint 1

	new profile for Professor.			
18	Perform acceptance tests for profile matching module.		7	Sprint 1
19	Integrate the front end and back	Integrate the front end and back end for - login, signup, home page, profile matcher module (all members need to validate their interfaces for correct working)	7	Sprint 1
20	Display the shortlisted students across the position		3	Sprint 2
21	Create a page to add questions/MCQ's/Link to test		3	Sprint 2
22	Link the test to shortlisted students		3	Sprint 2
23	Create a view for all TA positions		4	Sprint 2
24	Create a filter to filter out TA position based on - Subject, Time posted, Professor, Department, Domain etc.		4	Sprint 2
25	Enable the apply link only for eligible and active positions.		4	Sprint 2
26	Design front end for creating a ad about TA/RA positions.		6	Sprint 3
27	Design front end to apply to TA/RA positions.		5	Sprint 3
28	Implement controller for displaying/managing TA/RA positions.		6	Sprint 3
29	Write acceptance tests for a module that manages TA/RA positions		6	Sprint 3
30	Write a controller for applying to TA/RA positions and update database accordingly		5	Sprint 3

31	Write a persistence layer to update table for TA/RA positions		6	Sprint 3
32	Write acceptance tests for a module that allows students to apply for TA/RA		5	Sprint 3

Acceptance Criteria

User Story:

As a Student, I should be able to create a profile on TaRa so that I can access the Web Application.

1. Scenario:

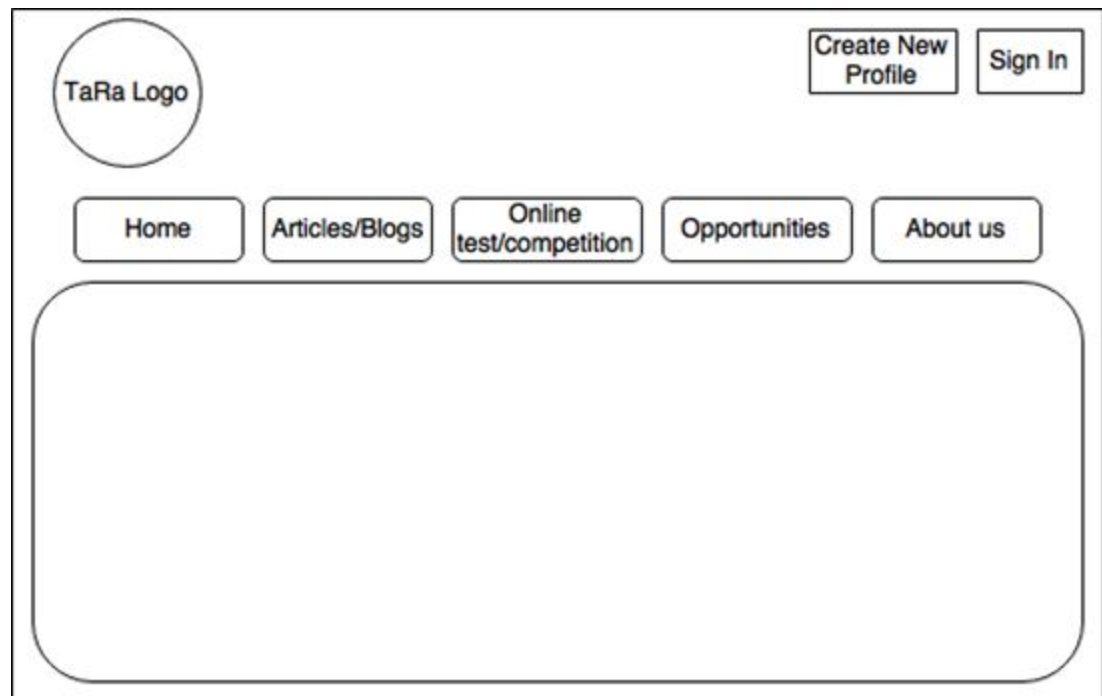
As a Student, I want to create a new profile on TaRa.

a. Acceptance Criteria:

[Student is directed to link for creating a new profile]

Given:

Student is on home page of TaRa web application. "Create a new profile" is in an activated mode and displayed on interface. Relevant entry for a student does not exist in database.




When:

Student clicks on "Create a new profile" tab on home page.

Then:

HTML form for creating a new profile will be popped out. "Create a new profile" tab will be disabled and will not be displayed on user interface.



Home

us


Name	<input type="text"/>
User ID	<input type="text"/>
Password	<input type="text"/>
Re-enter Password	<input type="text"/>
E-Mail ID	<input type="text"/>
User Type	Student ▼
Submit	

b. Acceptance Criteria:

[Student fills the form and successfully create a new profile]

Given:

Student is on HTML form for creating a new profile in TaRa web application. "Create a new profile" is in a deactivated mode and not displayed on user interface. Relevant entry for a student does not exist in database.



Home

us

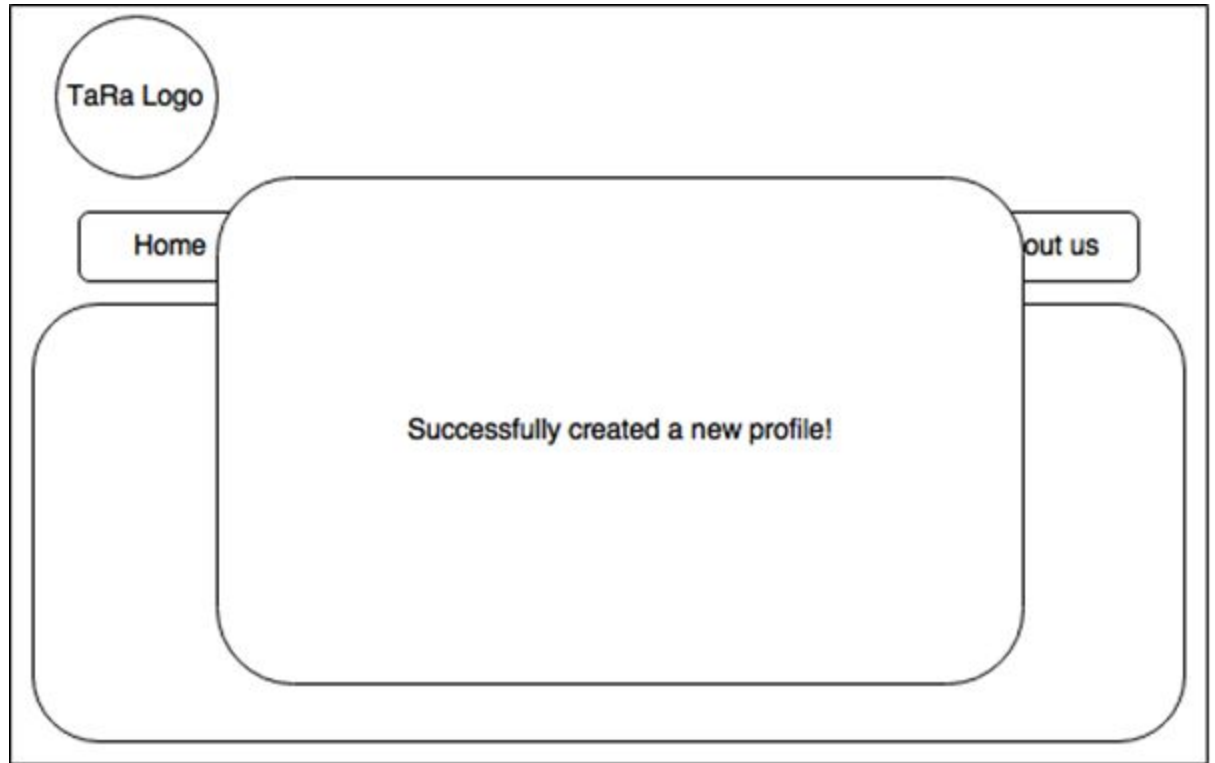
Name	<input type="text"/>
User ID	<input type="text"/>
Password	<input type="text"/>
Re-enter Password	<input type="text"/>
E-Mail ID	<input type="text"/>
User Type	Student ▼
Submit	

When:

Student fills the form and clicks on “Submit” button.

Then:

HTML form for successfully creating a new profile will be popped out. New user entry will be added into database.



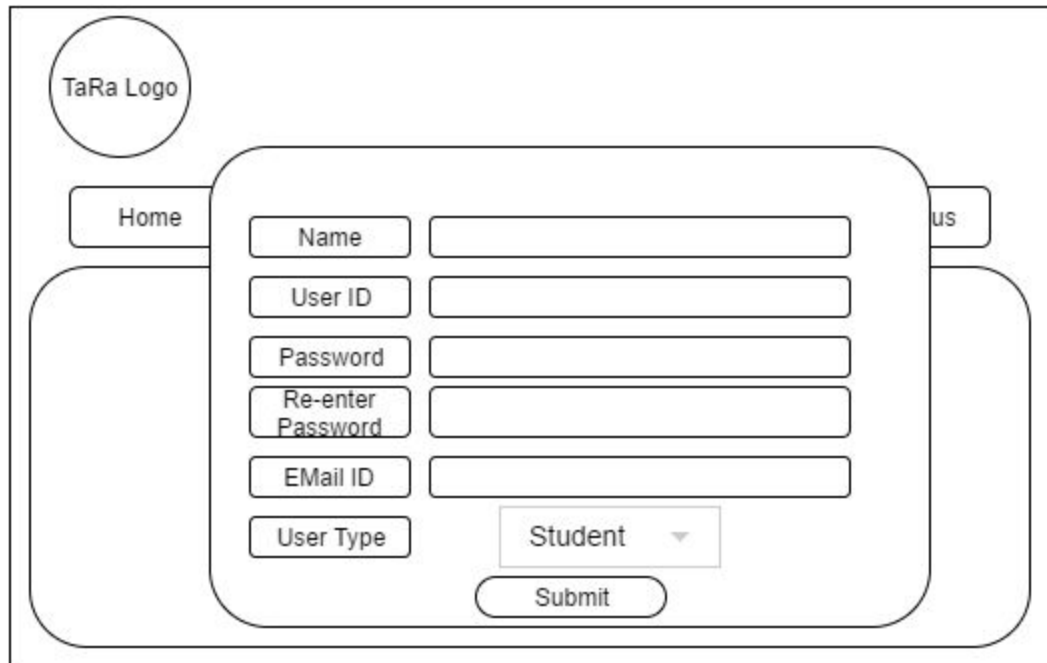
C. Acceptance Criteria:

[Student fills the form, but fails to create a new profile]

Given:

Student is on HTML form for creating a new profile in TaRa web application.

“Create a new profile” is in a deactivated mode and not displayed on user interface. Relevant entry for a student exists in database.



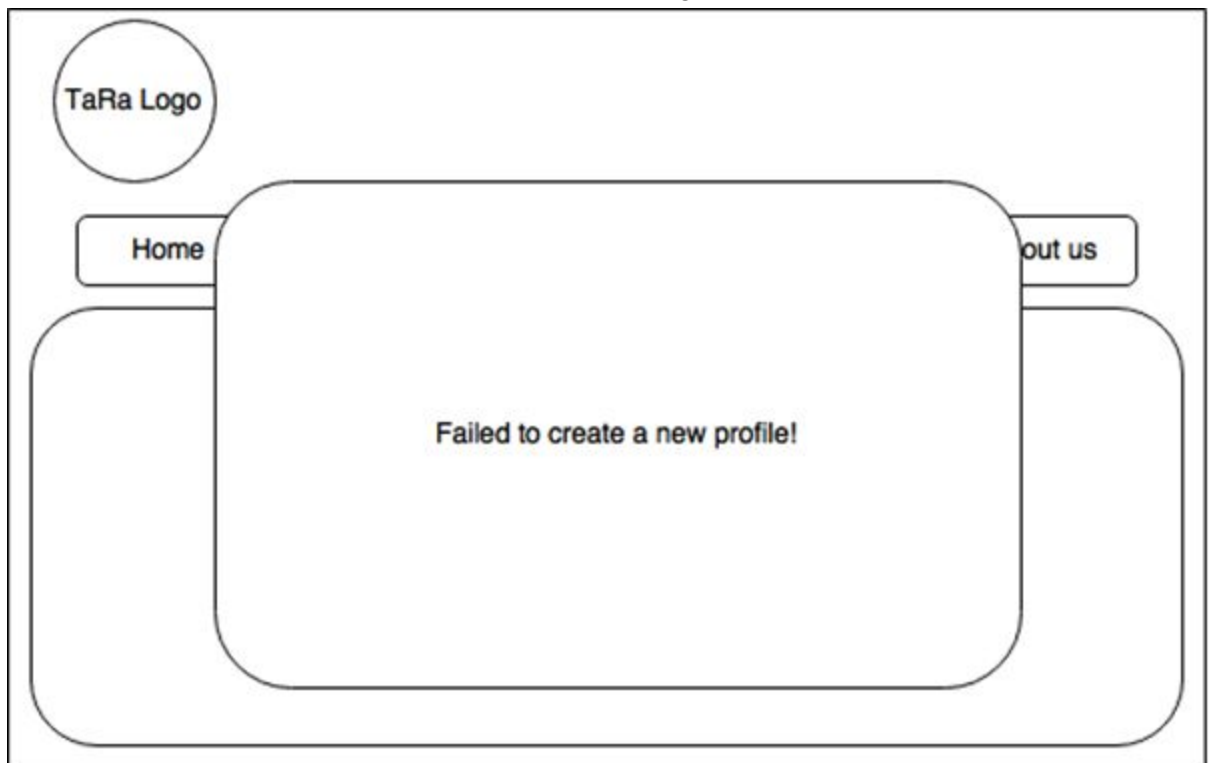
The image shows a user registration form within a web application interface. At the top left is the 'TaRa Logo'. Below it are navigation links for 'Home' and 'About us'. The form itself is a rounded rectangle containing several input fields: 'Name', 'User ID', 'Password', 'Re-enter Password', 'EMail ID', and 'User Type'. The 'User Type' field is a dropdown menu currently set to 'Student'. A 'Submit' button is located at the bottom right of the form.

When:

Student fills the form and clicks on "Submit" button.

Then:

HTML form for failing to create a new profile will be popped out. User entry will not be added into database and database is unchanged.



The image shows a message box that appears after a failed registration attempt. The message box is a rounded rectangle with the text 'Failed to create a new profile!' centered inside. It is overlaid on the same web application interface as the first image, showing the 'TaRa Logo', 'Home' link, and 'About us' link.

User Story:

As a Student I should be able to apply to open TA/RA positions so that my profile can be evaluated by the Professor.

1. Acceptance Criteria:

[Student clicks the apply button]

Given:

Student is on dashboard page of TaRa web application. With an account created

When:

Student clicks on "Apply" tab on dashboard page.

Then:

Application must be submitted for profile evaluation

User Story:

As a Professor, I should be able to manage the positions created by me, so that I can take necessary action.

1. Acceptance Criteria:

[Professor is presented with a form for creating a new position]

Given:

Professor is on dashboard page of TaRa web application. With an account created

When:

Professor clicks on "Create" tab on dashboard page.

Then:

Position form must be created

2. Acceptance Criteria:

[Professor is presented with a form for creating a new position]

Given:

Professor is on dashboard page of TaRa web application. With an account created

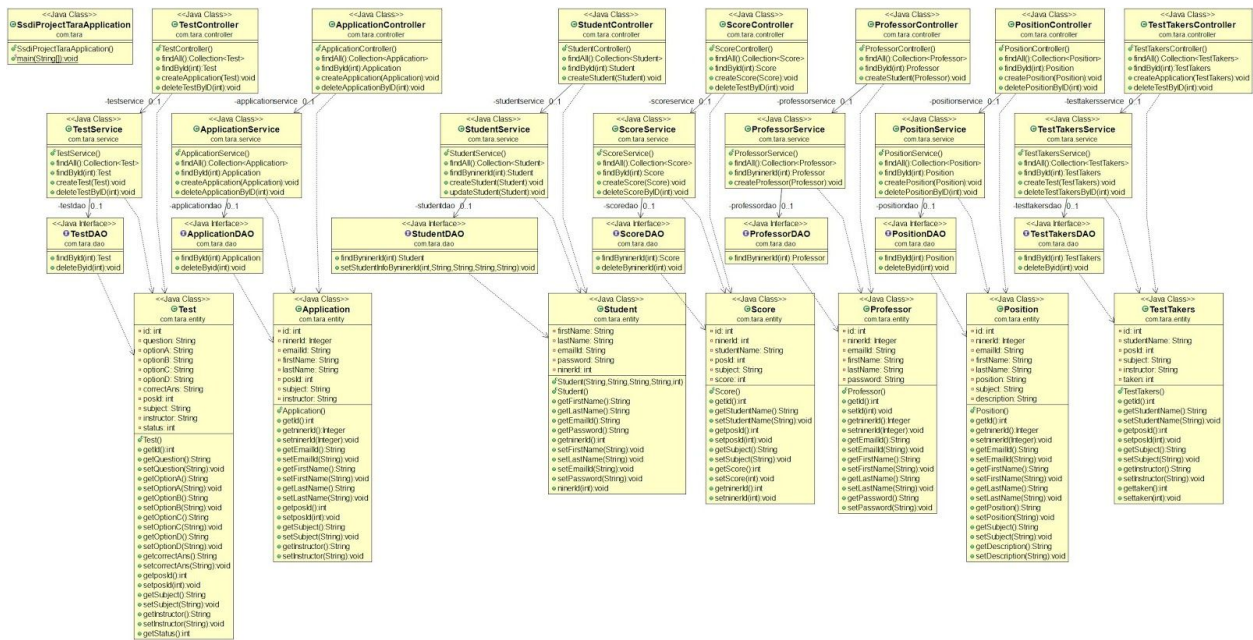
When:

Professor clicks on "delete" tab on dashboard page.

Then:

Position form must be created.

DETAILED DESIGN



TESTING

Testing is implemented using JUnit integrated with Spring Boot. The dependencies are managed by spring boot using beans. The test cases automated. The main focus is on unit testing and are done using separate packages.

SAMPLE TESTS

Test 3:

Requirement(s) under test: 2 - As a Professor, I should be able to create a profile on TaRa so that I can access the Web Application.

Function(s)/Module(s) under test: Professor

Initial conditions: Web-site is up and running

Assumptions: Sign-Up page is working

Test case input: All fields - First Name, Last Name, Niner Number, Email, Password are filled with valid values

Expected result: The professor account is created and database entry is made.

Result: Passed

Test 1:

Requirement(s) under test: 1- As a Student, I should be able to create a profile on TaRa so that I can access the Web Application.

Function(s)/Module(s) under test: Student

Initial conditions: Email Address is left empty while signing up

Assumptions: Sign-Up page is working

Test case input: All fields - First Name, Last Name, Niner Number, Password are filled with valid values.

Expected result: The student account is not created and error is thrown on the screen.

Result: Passed

Test 7:

Requirement(s) under test: 5- As a Student I should be able to apply to open TA/RA positions so that my profile can be evaluated by the Professor

Function(s)/Module(s) under test: Application

Initial conditions: A student is logged in and there is one active position to apply

Assumptions: Student logs in to the account

Test case input: A student applies to present active position

Expected result: The application is submitted and database entry is made

Result: Passed