

# CRITERION A- PLANNING

## DEFINING THE PROBLEM

My client is a teacher, the activity manager of the school, who is in-charge of the events, competitions, and shows that takes place in school or that involves the participation of the school, such as Flair-fest and MUN. In consultation with the client<sup>1</sup>, she stated that currently she has to ask someone to make the announcements or spread a word, or sometimes, she has to do it on herself to make sure that most of the students get to know about the event or competition. Also, for some events, mail is being sent. For participation, students need to again-and-again approach the teacher (client) or if they are able to view the mail, they register themselves. Further, it gets uncomfortable and incompetent for both the teachers and students to approach each other. This involves a lot of physical efforts which becomes hectic for the client as well as the students. Involving a lot of physical efforts results in inefficiency. Plus, it is a headache for the teacher to go and look for students who are willing to participate. My client does not want to miss out on anyone's talent. Many students miss the opportunity to participate as they don't check their mails regularly and are not aware of it. Sometimes, students are not able to access their mails and so, they miss the chance to register for the events. They also often register after the deadline is crossed as they aren't aware of it before. And, finalizing the students are done manually on a piece of paper which sometimes might go missing. To solve these problems, I have decided to make a mobile app which would handle all the events and the registrations.

**Word count – 284**

## RATIONALE FOR SLUTION

---

<sup>1</sup> Refer to Appendix 1

I will be making a mobile app that would solve the problems mentioned above and make the process of registering for events easier<sup>2</sup>. Considering that we aren't available on our laptops/computers all the time, a mobile app would be more efficient as mobile phones are used on a regular basis. I will be using java script programming language for the client interface. For the front-end, I am using react native as it makes the development phase faster and it has a fast learning curve. It has a hot reloading feature which enables the function to be saved on the app as soon as it is typed as a code. Moreover, it has a very strong library which can be used for school event management system. In the UI designing process, I will be using FLEX which uses CSS (Cascading Style Sheets). The teacher can add events for the students to register for the particular event on the app. On registering, the teacher can view the details of the students who have registered. By this, the interaction of end users, that is, teachers and students become easier, and no more physical efforts have to be put in by both the students and the teacher. To finalize the students for an event, the teacher can either approve or disapprove a student on the app itself. There will be one admin who will already be signed-up and can add or remove teachers. As the app will require sending mails to user on their log ins and student's selection for the event, a server is needed. I will be obtaining my server/cloud service from DigitalOcean. MYSQL will be used for the database, and the backend programming language I will be using is PHP. Apache will be my backend server.

**Word count – 300**

## **SUCCESS CRITERIA**

1. Different log-ins and sign-ups for students, teachers, and admin.
2. Security for teacher signup: only the added teachers are being able to sign-up. Admin can add or remove teachers.
3. OTP verification on signing-up.
4. Different home screen functionalities for teachers, students, and admin.

---

<sup>2</sup> Refer to Appendix 1

5. On admin log-in,

- 5.1. Three buttons available: 'Add Teacher', 'View Teachers', and 'View Admin'.
- 5.2. 'Add Teacher' lets the admin add a teacher by entering his/her e-mail and making the signing up available for the new teacher added.
- 5.3. 'View Teachers' directs the admin to a screen where a list of all signed-up teachers is displayed. Clicking on a teacher from the list will showcase the details of the teacher, entered while signing-up, and below the details will be a 'Delete' button.
- 5.4. On clicking 'Delete', the teacher will be removed from the list and its data will be erased from the database.
- 5.5. 'View Admin' directs the user of a screen where the details of the admin are displayed. Bottom of the screen will have a 'Delete' button.
- 5.6. Clicking the 'Delete' button will direct the user to the 'Add Admin' screen.
- 5.7. Clicking the 'Add' button will add the new admin registered and remove the previous admin. The log-in credentials of the new admin will be sent on the mail of the new admin.

6. On teacher log in,

- 6.1. 'Add Event' lets the teacher to add an event. Adding an event makes the event viewable on the home screen for both students and teachers.
- 6.2. The button, 'Delete', on the events card will delete the event and remove it from the screen.
- 6.3. Clicking on an event, on the home screen, should showcase the list of students registered for the event.
- 6.4. Clicking on a student in the list showcased should display the details filled by the student on the registration form. At the bottom of this screen are 2 buttons: approve and disapprove.
- 6.5. On clicking 'Approve', the student should remain on the list of registered students. A mail should be sent notifying the student's selection for the event.

On clicking 'Disapprove', the student is removed from the list of registered students. A mail should be sent notifying the student's refusal for the event.

7. On student log-in,
  - 7.1. events that are added by the teacher are displayed on the home screen.
  - 7.2. clicking on an event makes a registration form available to the student, where the student can register for the event.
  - 7.3. 'submit' button enables the form to be submitted and the student to be registered if all the fields are filled appropriately. A message saying "event registered successfully" will be shown.
  - 7.4. A student cannot fill a particular event form twice.
8. 'Forgot Password' function lets the users receive their password again by sending them the password on their mail.
9. All the inputs while signing up, logging in, filling form, adding event, and adding teacher are mandatory. An error message to be shown if any of the input fields are left blank.