

# CRITERION B- DESIGN

As mentioned in Criteria A, I will be designing an app that would solve the problems<sup>1</sup> mentioned in Criteria A and make the process of registering for events easier for students and the process of handling the events easier for teachers. I would be developing my app using JavaScript programming language for the client interface. For the front-end, I will be using react native, and in the UI designing process, I will be using FLEX which uses CSS (Cascading Style Sheet). I have planned to design my app being specific to what my client wants and trying to make it as user friendly as possible. My app will be having these main screens:

- 1) Log-in Screen
- 2) Teacher Sign-up Screen
- 3) Student Sign-up Screen
- 4) Events Screen (Home Screen for Students, Teachers, and Admins)
- 5) Add Teacher Screen (for the admin to add another teacher)
- 6) View Teachers Screen (for the admin to view the signed-up teachers)
- 7) Teacher details screen (for the admin)
- 8) View Admin Screen (for the admin)
- 9) Add Admin Screen (for the admin)
- 10) Add Event screen (for the teacher to add a new event)
- 11) Student list Screen (for the teacher to view the students those have registered for the event)
- 12) Student Form's Details Screen (for the teacher to view what the student has filled on the form)
- 13) Registration Form Screen (for the student)

---

<sup>1</sup> Refer to Criteria A.

## 1<sup>st</sup> screen – LOG-IN SCREEN:

Log-in Screen :-

Name of the app. School's logo

INPUT ← ENTER E-MAIL

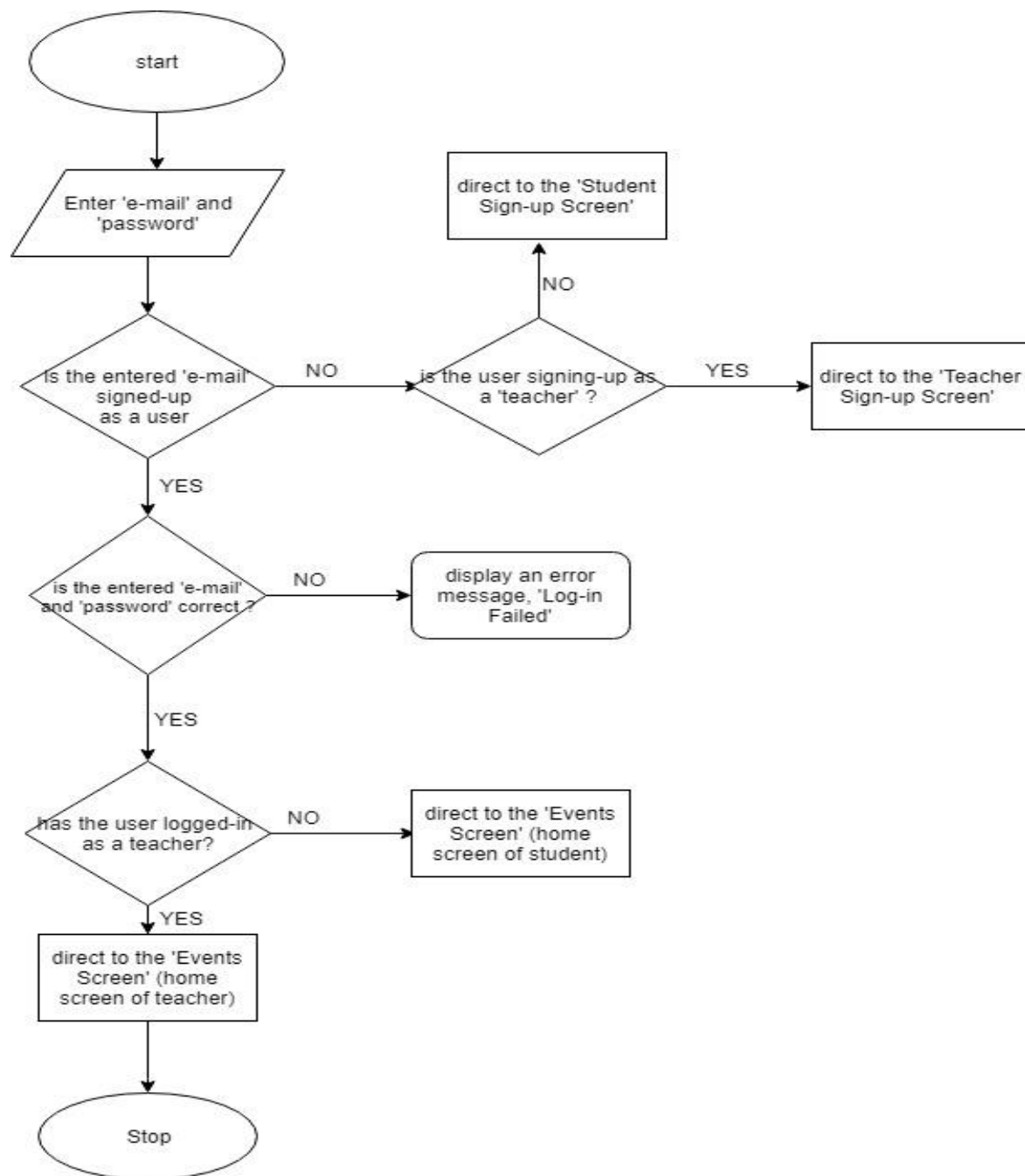
INPUT ← ENTER PASSWORD

BUTTON ← LOG-IN Teacher LOG-IN student → BUTTON

Forgot Password? → BUTTON

Don't have an account? SIGN-UP → BUTTON

Above, is a layout/design of the Log-in Screen I have thought of. This will be the default screen of the app if the user is logged out, therefore this is the screen the user has to go through on his/her first time accessing the app. The user then has to sign-up by clicking the 'Sign-up' button which will give user the option of either signing-up as a teacher or as a student. For security purposes, an OTP will be sent to the email that is used for signing-up to verify the user. If the user is already signed-up, then the user has to enter his/her 'e-mail' and 'password' to log-in as a teacher or student respectively. The app will then check if the user is classified as a teacher, student, or admin as per the user's sign-up details. Therefore, if a student tries to log-in via 'Teacher log-in', and vice-versa, then there will be an error in the log-in process. If the user forgets his/her password, then the user may click the 'Forgot Password' button and receive an OTP on the email that the user had signed-up with. On entering the correct OTP, a mail will be sent to the user with the user's password. Below is a flowchart representative of the functioning of this screen.



## SIGN-UP SCREENS (TEACHER AND STUDENT):

Sign-up Screen (Student) :-

Logo

Student Sign-up

FULLNAME	→ INPUT
E-MAIL	→ INPUT
PHONE NO.	→ INPUT
CLASS	→ INPUT
SECTION	→ INPUT
PASSWORD	→ INPUT
CONFIRM PASSWORD	→ INPUT

BUTTON ← Sign-up

Scanned with CamScanner

Sign-up Screen (Teacher) :-

Logo

Teacher Sign-up

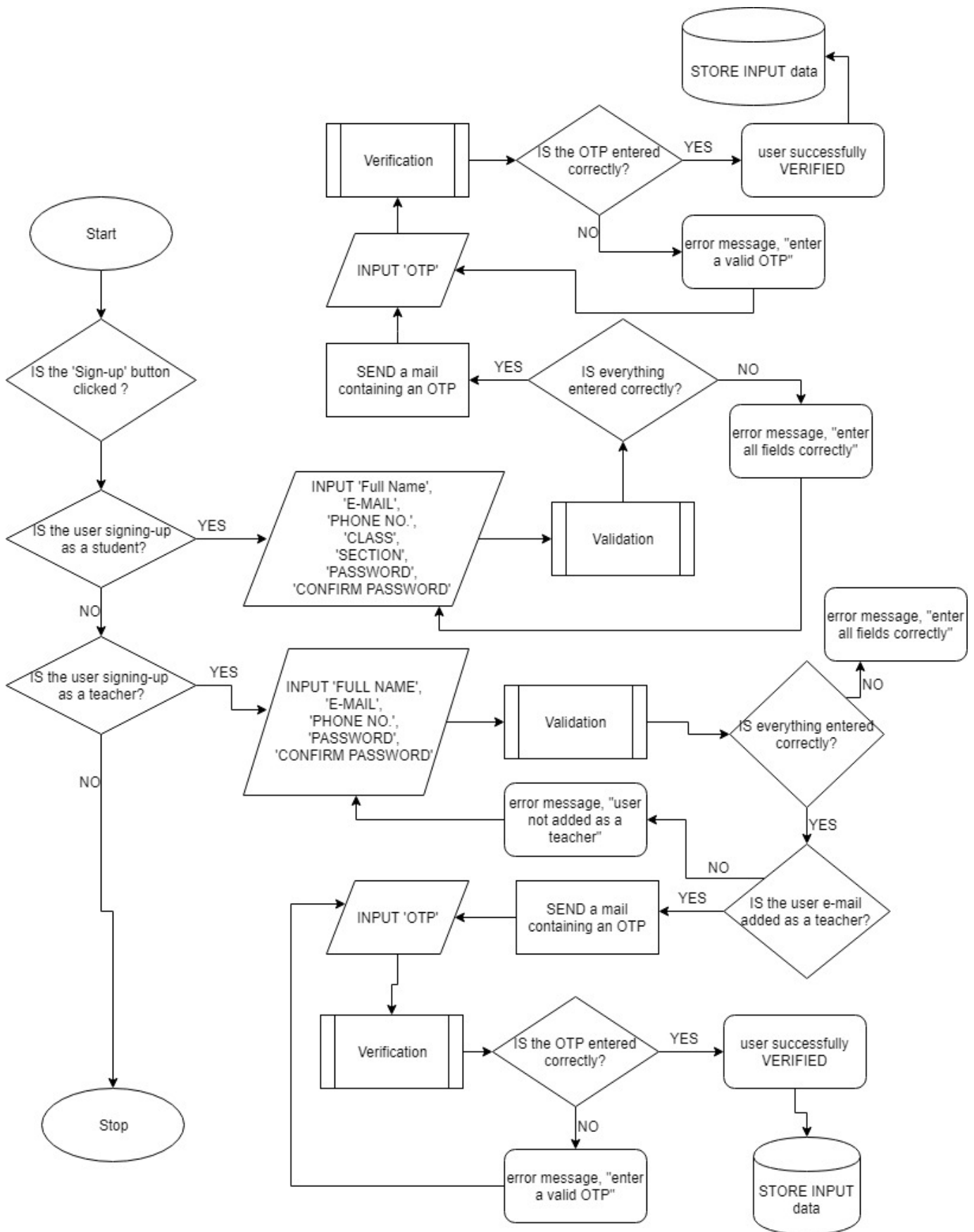
FULLNAME	→ INPUT
E-MAIL	→ INPUT
PHONE NO.	→ INPUT
PASSWORD	→ INPUT
CONFIRM PASSWORD	→ INPUT

BUTTON ← Sign-up

Scanned with CamScanner

There will be 2 sign-up screens: one for the teacher and the other for the student. If the user clicks on the 'Sign-up' button on the 'Log-in' screen, then the user will be provided with these 2 options to sign-up. If the user chooses to sign-up as student, after entering the information that is required, an OTP will be sent to the e-mail the user has used to sign-up with. On accessing the e-mail and checking the OTP, if the user enters the correct OTP then the user is successfully verified as a student and can now log-in. A user can only sign-up as a teacher if the admin has added the user as a teacher using the 'Add Teacher' button on the admin's home screen. If the new teacher's e-mail is added, the user can then sign-up as a teacher. On entering the e-mail to sign-up, there will be a check to see if the e-mail is added by any teachers. Then the user has to go through the same process of OTP to verify themselves<sup>2</sup>.

<sup>2</sup> Criterion A, Success criteria 3



### HOME SCREEN (Events Screen for Admin):

Events Screen (Home Screen) :-

ADD TEACHER VIEW TEACHER

VIEW ADMIN

EVENT NAME

Description: Fee: Age group venue: Start Date: End Date:

an event card

EVENT NAME

Description: Fee: Age: another event

For Admin (on admin login)

This screen is the home screen for the admin. Events that are added by the teacher are displayed on this screen as event cards. Three buttons are available to the admin following three different functions, 'Add Teacher', 'View Teachers', 'View Admin'.<sup>3</sup>

### ADD TEACHER SCREEN (Teacher):

ADD TEACHER SCREEN :-

School logo

ADD TEACHER

Teacher Name

E-mail

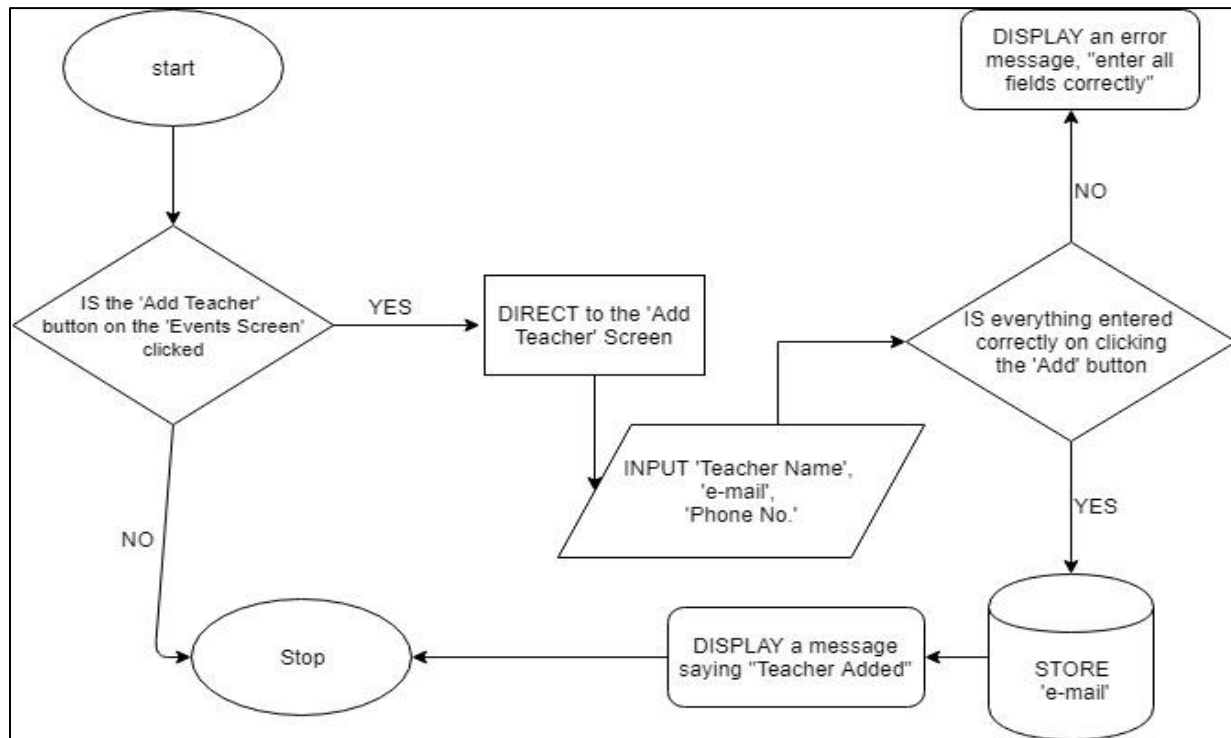
Phone no.

ADD

CS Scanned with CamScanner

<sup>3</sup> Appendix 2 and Criterion A, Success criteria 5.

The 'Add Teacher' function is for further security purposes. By clicking the 'Add Teacher' button on the events screen, the admin will be directed to this screen. By entering an email of the teacher that has to be added and clicking the 'Add' button, a new teacher will be stored. Only now can a new teacher sign-up (on the 'Teacher Sign-up Screen') with the email that was entered by the admin on this screen.

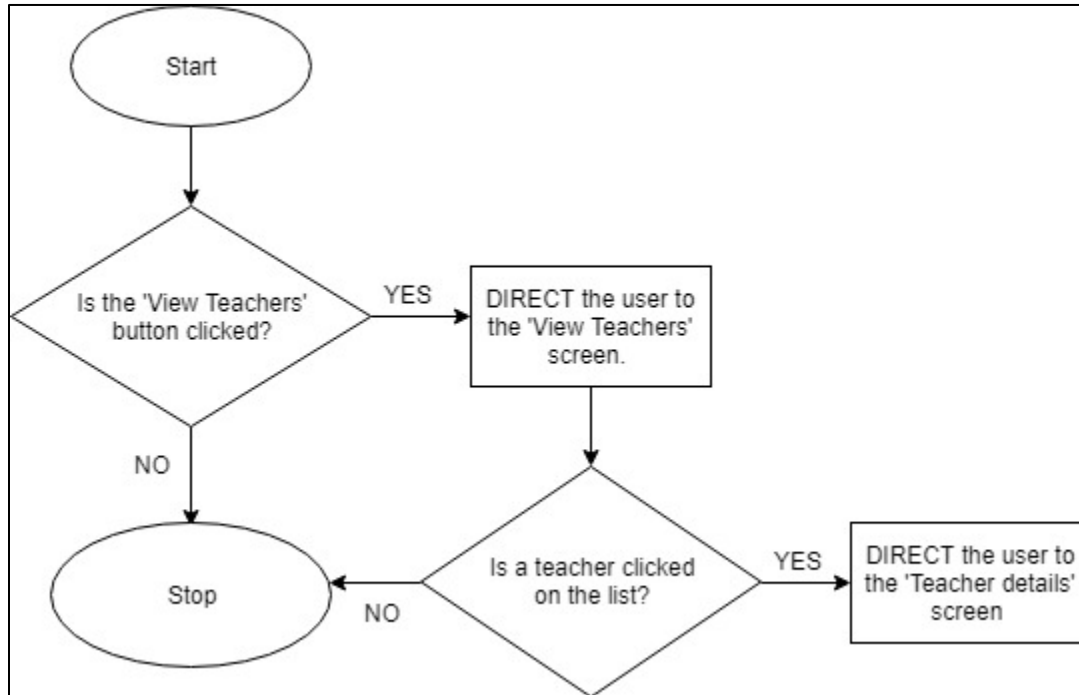


### VIEW TEACHERS SCREEN:

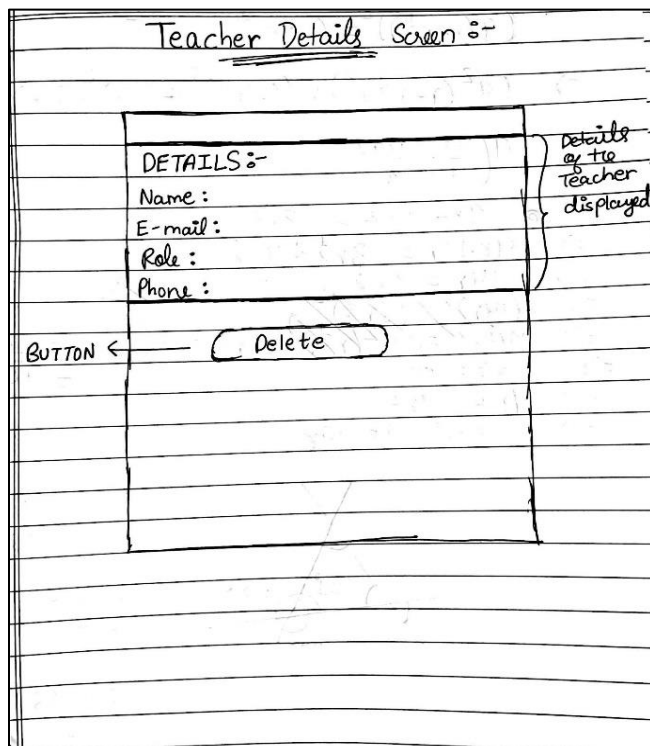
View Teachers Screen:-

	Teacher's Name	
Sample data	Minal Jain	} list of Teachers log sign up.
Division between teachers		

This screen is accessible from the admin's home screen. On clicking the 'View Teachers' button, the user will be directed to this screen where a list of signed-up teachers is displayed. With this screen, admin will have a track of all the teachers registered on the app. Clicking on one of the teachers on the list will direct the user to the 'Teacher details' screen.

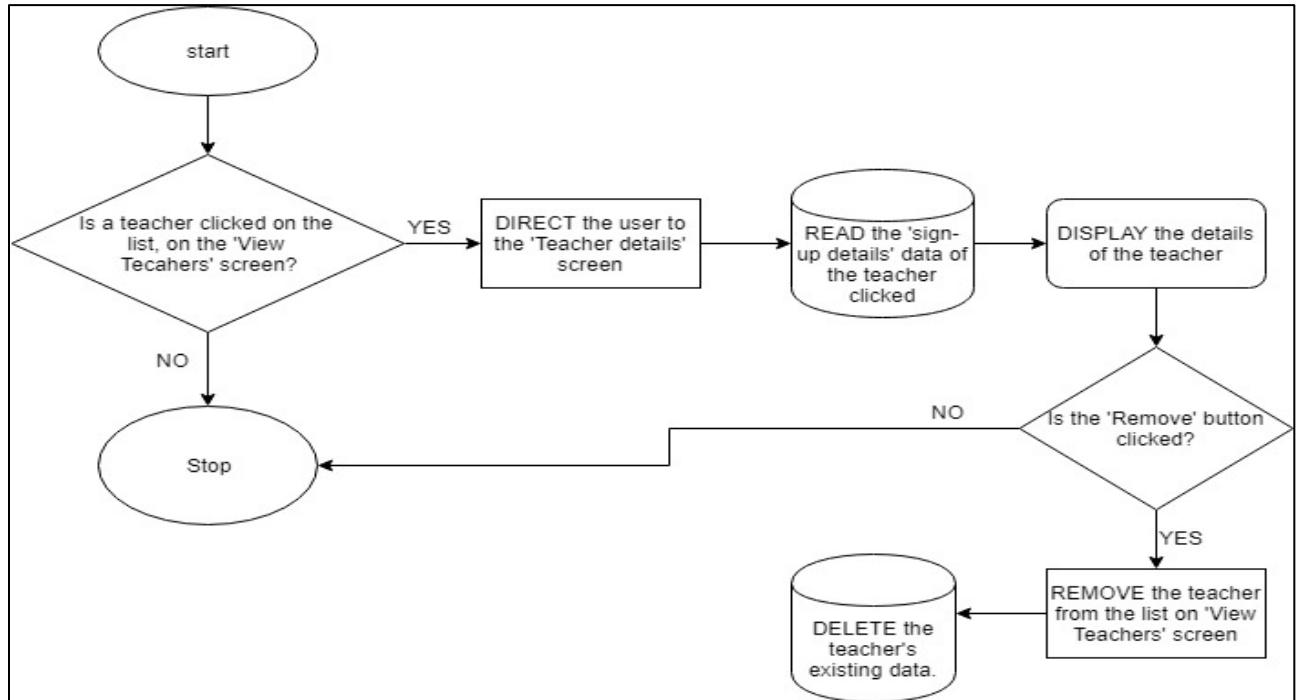


#### TEACHER DETAILS SCREEN:





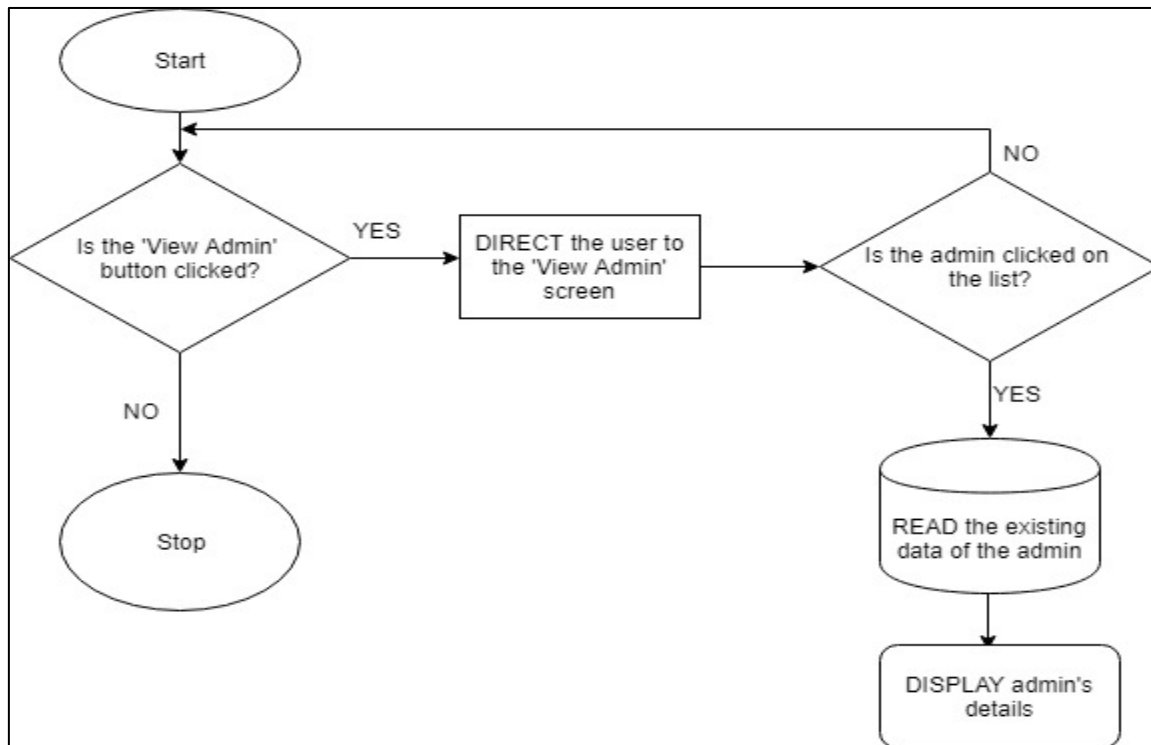
This screen is obtained by clicking on a teacher from the list on 'View Teachers' screen. This screen displays the details of the teacher clicked on the 'View Teachers' screen. Below the details of the teacher, there is a 'Remove' button. On clicking that button, the teacher is removed in case, for example, if a teacher leaves the school. The teacher is then removed from the list and all the existing data of that teacher is also erased.



### VIEW ADMIN SCREEN:

The sketch shows a rectangular window titled 'View Admin Screen:'. Inside the window, there is a section labeled 'DETAILS:-' containing four lines of text: 'Name:', 'E-mail:', 'Role:', and 'Phone:'. To the right of these details, there is a vertical bracket and the text 'Details of the admin displayed'. Below the details section, there is a label 'BUTTON ←' pointing to a rounded rectangular button labeled 'Delete'.

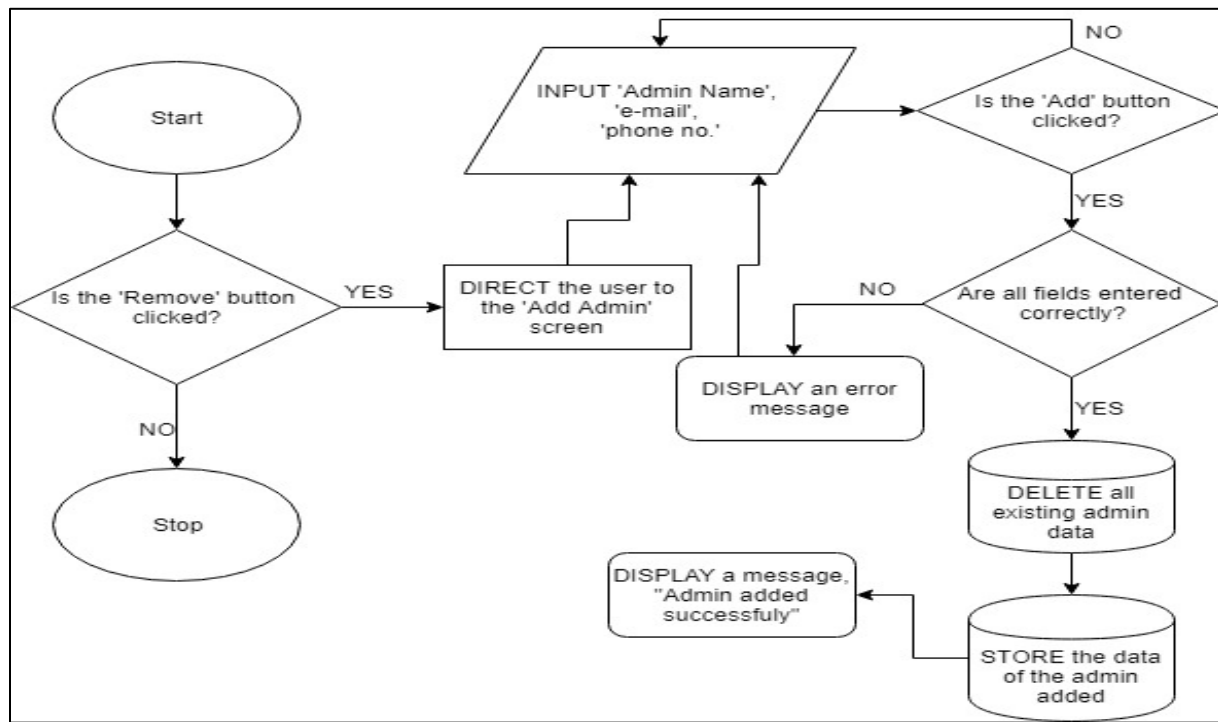
The user is directed to this screen by clicking the 'View Admin' button from the admin's home screen. 'View Admin' button will display a screen where the name of the current admin will be shown. This screen will appear on clicking that name, and will have the details of the admin. Removing and adding an admin is done by the 'Delete' button at the bottom of the details.



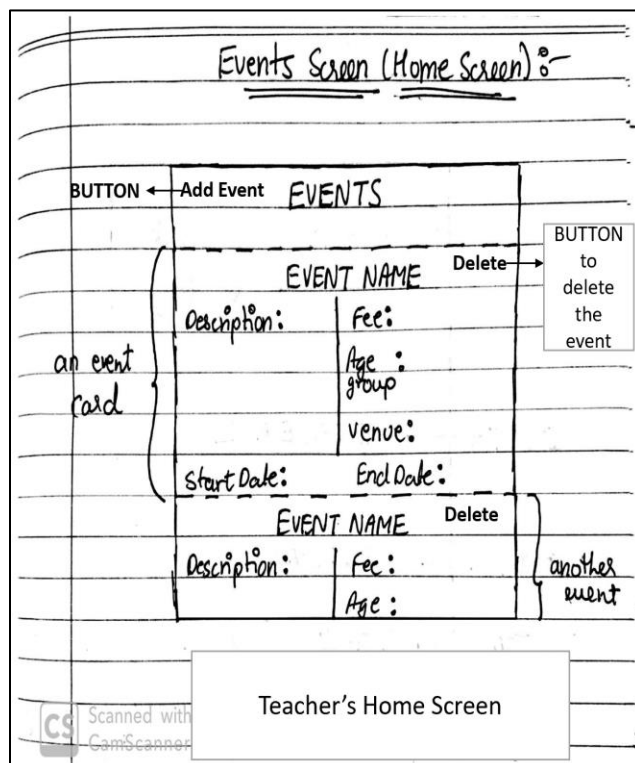
#### ADD ADMIN SCREEN:

The image shows a hand-drawn user interface for an 'Add Admin Screen'. At the top, there is a title box labeled 'Add Admin Screen'. Below it is a rounded rectangle containing a 'School logo' and an 'ADD ADMIN' button. Underneath the rectangle are three input fields labeled 'Admin Name', 'E-mail', and 'Phone no.'. To the left of these fields, there are three 'INPUT' labels with arrows pointing to each field. Below the input fields is a button labeled 'ADD', with a 'BUTTON' label and an arrow pointing to it. The entire design is drawn on lined paper. A watermark at the bottom left reads 'CS Scanned with CamScanner'.

This screen is obtained by clicking the 'Delete' button on the previous screen, 'View Admin' screen. Clicking the button will direct the user to the 'Add Admin' screen and once the admin is added successfully, the previous admin is automatically removed.



### HOME SCREEN (Events Screen for Teachers):



This screen is the home screen for teachers. Events that are added by the teacher is displayed in an event card form. 'Add Event' function is available to the teacher for adding new events. Students can then register for the events added by the teacher. Several functions are also performed by clicking on an event card.<sup>4</sup>

#### ADD EVENT SCREEN (Teacher):

Add Event Screen :-

School  
logo

ADD EVENT

EVENT NAME	→ INPUT
DESCRIPTION	→ INPUT
FEE	→ INPUT
CLASSES	→ INPUT
VENUE	→ INPUT
START DATE	→ INPUT
ENDDATE	→ INPUT

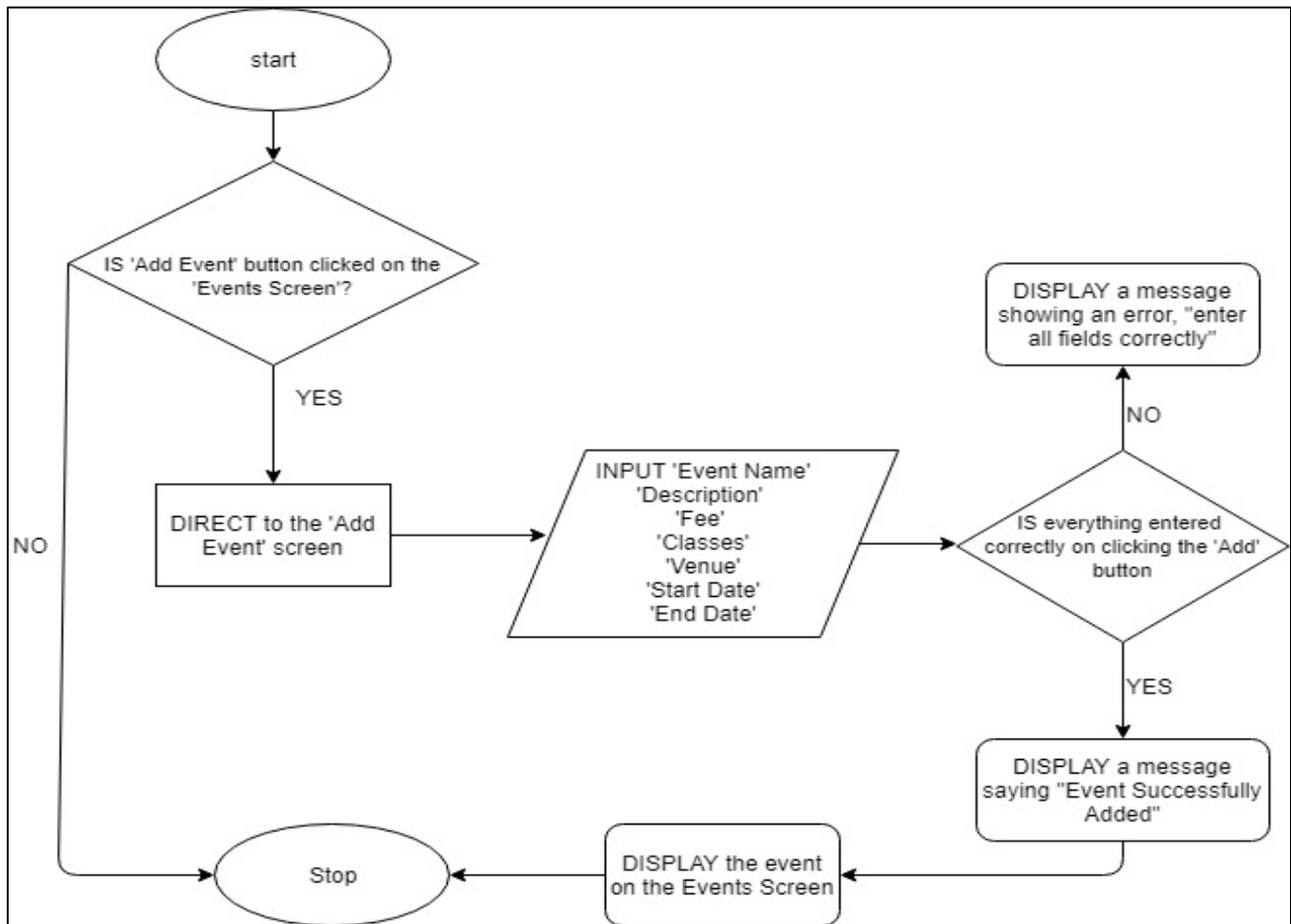
BUTTON ←

ADD

Scanned with CamScanner

This is the layout of the screen which requires certain details of the event as an input. On clicking the 'Add' button, an event will be added and is made visible to both the teachers and the students on the 'Events Screen' which is the home screen. This is a function that only the teacher can use.

<sup>4</sup> Criterion A, success criteria 6.



### 3<sup>rd</sup> Screen (Teacher) – STUDENT LIST SCREEN:

Student List Screen :-

Student Name	Class Section
Viraj Vijaywargiya	12 A

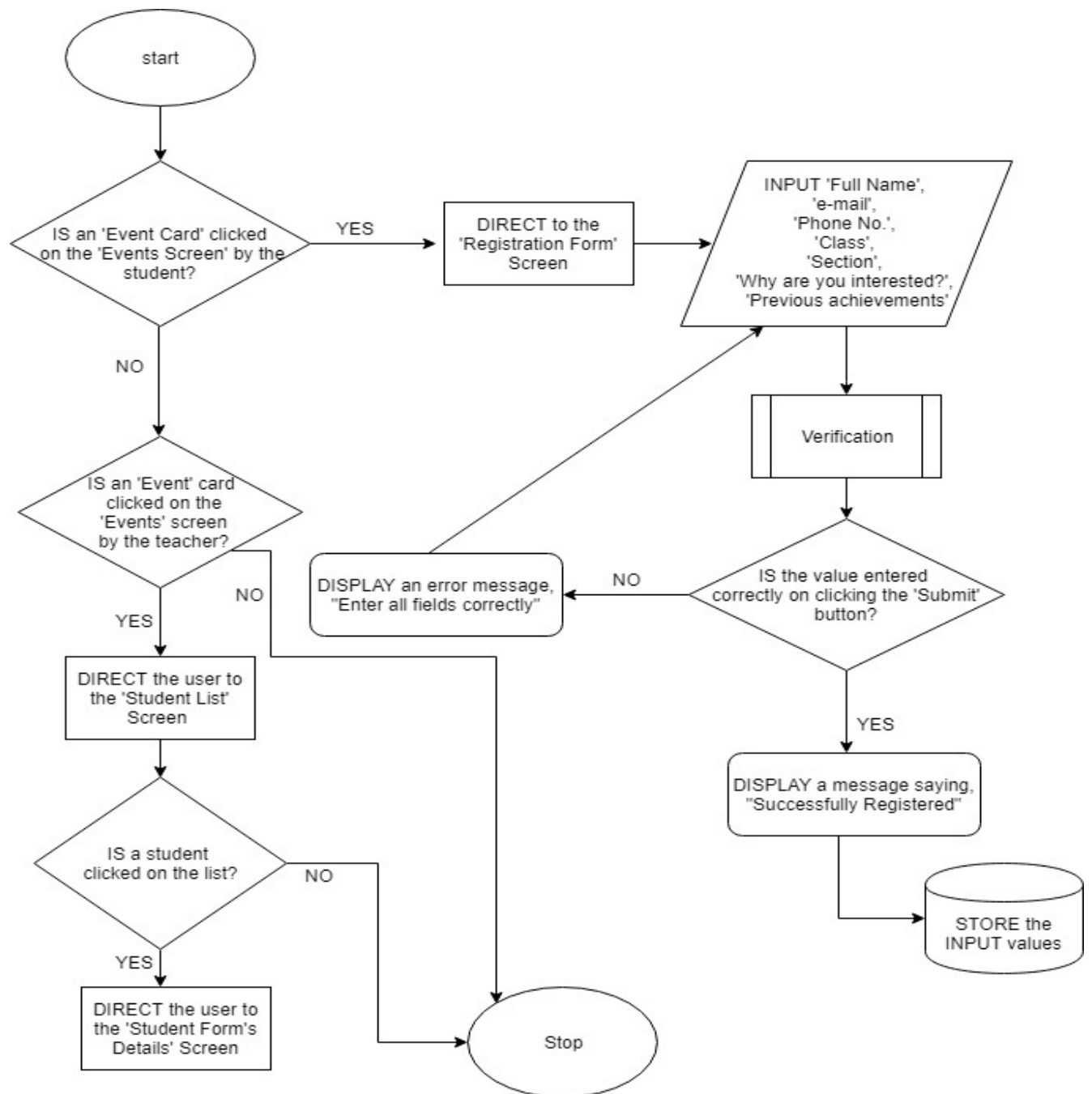
Sample Data ← division & blw students

list of students registered

Scanned with CamScanner

This is a screen that is accessible on teacher's log-in. On clicking the 'Event card' on the 'Events Screen', the teacher will be directed to this screen which contains a list of all the students who

have registered for the event clicked. By this, the teacher can have a digital form of the list of students who are willing to participate. However, it is later on the teacher if he/she wants to 'approve' the student or 'disapprove' the student for the event. Below is a flowchart representative of this screen. The flowchart below also contains functionality for student on clicking an event card. This screen is the screen after the home screen (Events Screen).



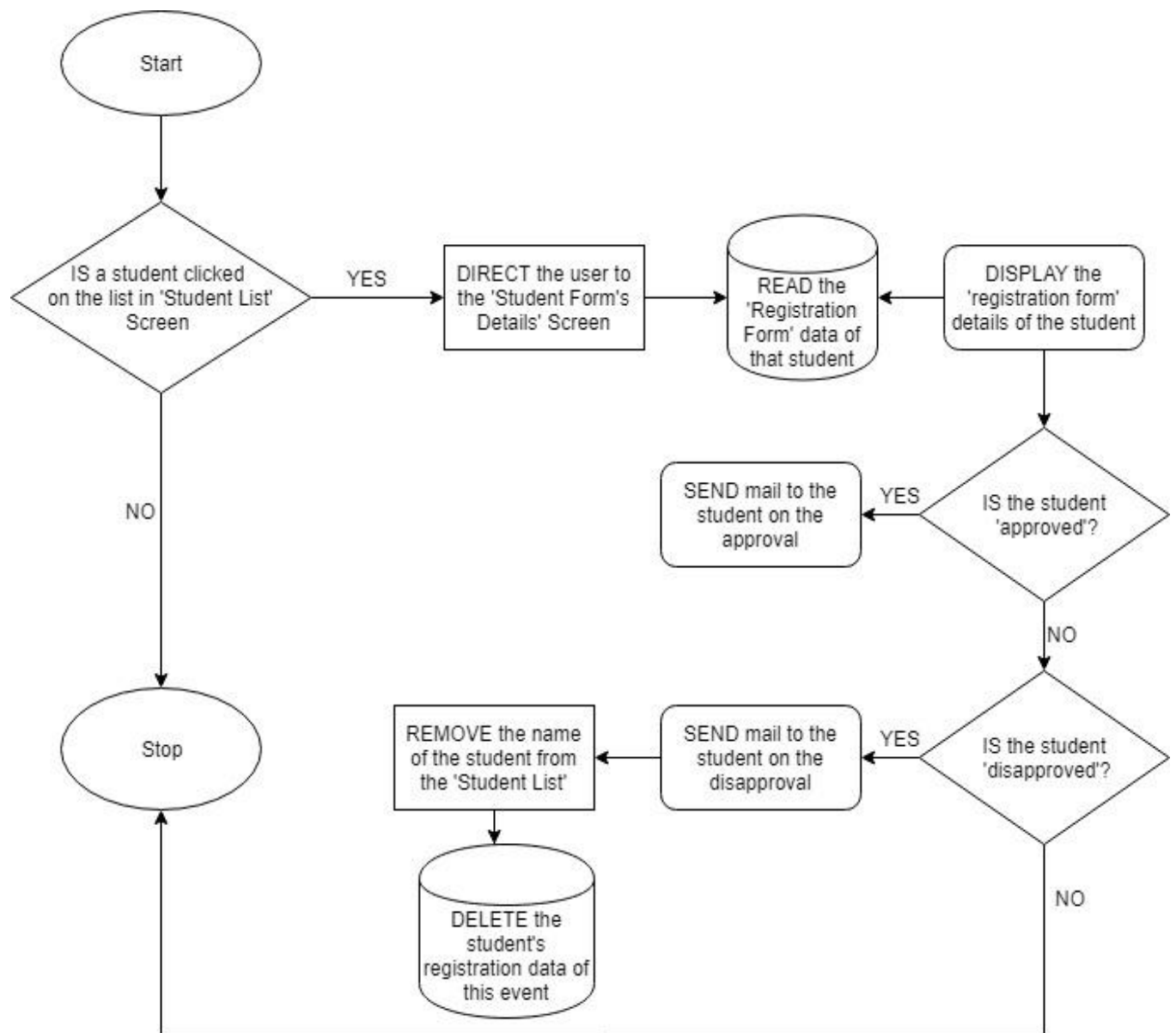
## STUDENT FORM'S DETAILS SCREEN:

Student Form's Details Screen :-

EVENT NAME	
Details :	
FORM	
FULL NAME	→ DISPLAY
E-MAIL	→ DISPLAY
PHONE NO.	→ DISPLAY
CLASS	→ DISPLAY
SECTION	→ DISPLAY
WHY ARE YOU INTERESTED?	→ DISPLAY
PREVIOUS ACHIEVEMENTS	→ DISPLAY
BUTTON ← <input type="button" value="Approve"/> <input type="button" value="Disapprove"/> → BUTTON	

CS Scanned with CamScanner

This is the design of the screen that displays the details the student has filled in the registration form. The screen has the same design as the 'Registration Form' Screen, except for the fact that this screen is a display form of it and contains an additional 'approve' and 'disapprove' function. On clicking a student on the list in the 'Student List' Screen, this screen will show up displaying that student's details according to the form filled. By looking at the details, if a teacher feels that the student is not eligible for the event, the teacher can 'disapprove' the student. By doing this, the student's name and details will be removed from the list of students on the 'Student List' screen. If a student is 'approved', then the student will remain on the list.





## HOME SCREEN (Events Screen for Students):

Events Screen (Home Screen) :-

EVENTS	
EVENT NAME	
Description:	Fee:
	Age : group
	Venue:
Start Date:	End Date:
EVENT NAME	
Description:	Fee:
	Age :

(For Students) on student log-in

This screen is the home screen for students. All the events added by the teacher are displayed on this screen in form of event cards. A student can fill the registration form of an event by clicking on an event card.<sup>5</sup>

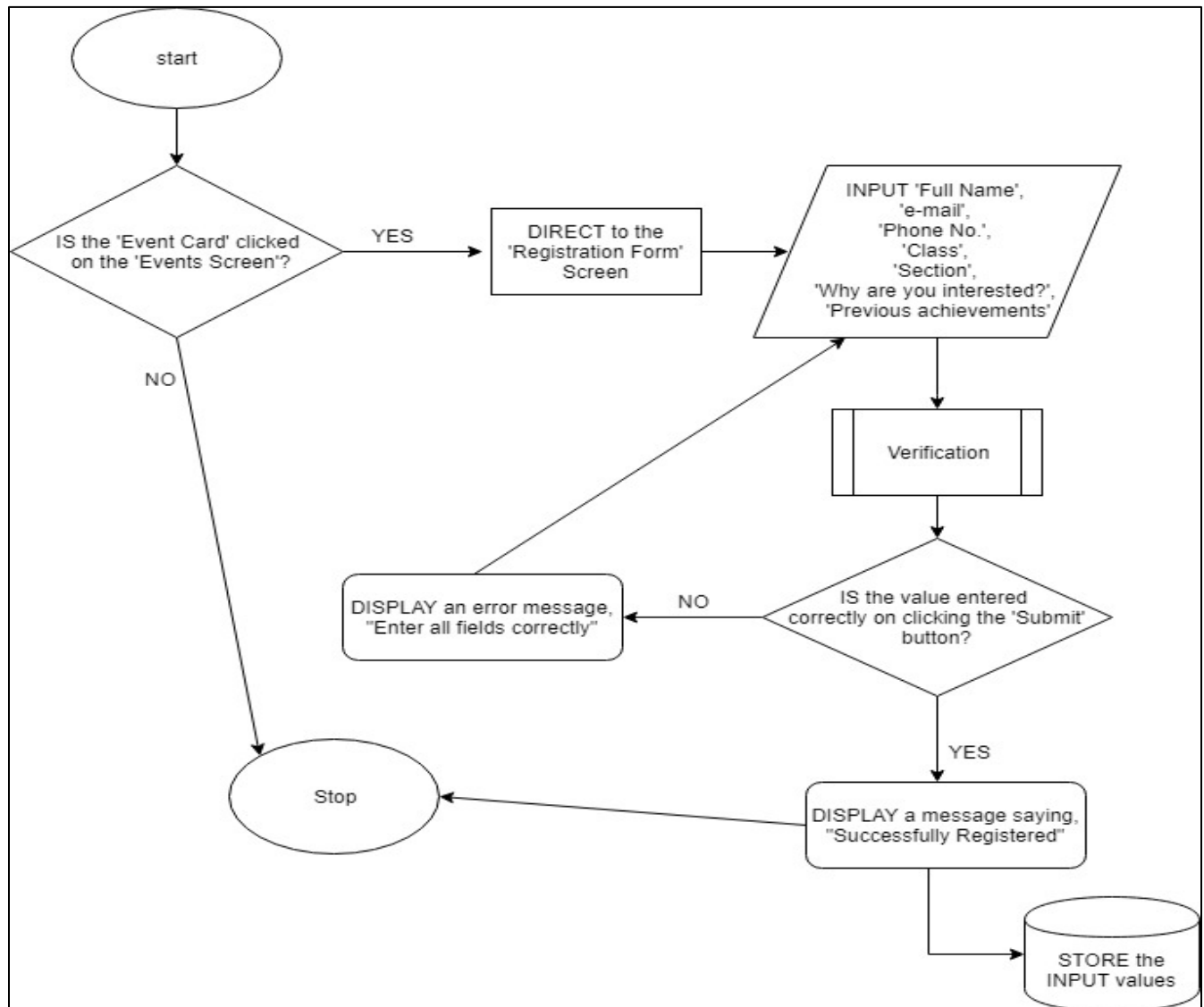
## REGISTRATION FORM SCREEN:

Registration Form Screens :-

EVENT NAME	
Details:	
FORM	
FULL NAME	→ INPUT
E-MAIL	→ INPUT
PHONE NO.	→ INPUT
CLASS	→ INPUT
SECTION	→ INPUT
WHY ARE YOU INTERESTED?	→ INPUT
PREVIOUS ACHIEVEMENTS	→ INPUT
BUTTON ← (SUBMIT)	

<sup>5</sup> Criterion A, success criteria 7.

This screen is designed for the students to fill a form to register for the event. By clicking an event card on the events screen (home screen for the students), the user will be directed to this screen that contains the registration form for the event selected. After the user has filled the form, clicking on the 'Submit' button will give out a message saying 'Successfully Registered'. By this the user will know that he/she has registered successfully. The information that is filled in the form can be viewed on the 'Student Form's Details Screen', where the teacher will be able to view the details filled on the form.



## **BACK-END:**

For the database, I will be using MYSQL. The back-end programming language I will be using is PHP. For 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. Apache will be my backend server. I will not require any other table to store the registration details or the teacher details as all this data will be obtained from the 'USER' table itself.

### **TABLES:**

#### 1) USER

- Name
- E-mail
- Class
- Section
- Phone No.
- Password
- Type ('1' for students, '2' for teachers, and '3' for admin)

#### 2) EVENTS

- Title
- Description
- Fee
- Age-Group
- Venue
- Start Date
- End Date

**TEST TABLE:**

S/NO.	TEST	SAMPLE DATA	EXPECTED OUTCOME
1	E-mail Verification on signing-up	OTP sent to the user's e-mail to verify the user. 'OTP: 2345'	If the OTP is correct, display a successful sign-up message. If the OTP is incorrect, display an error message.
2	Login (True)	Email - <a href="mailto:virajvijaywargiya@yahoo.com">virajvijaywargiya@yahoo.com</a> Password - 12345	User directed to home screen if login is successful.
3	Teacher sign-up verification	Check if the entered e-mail is existing in the database. 'xyz@gmail.com'	If the e-mail is existing in the database, signing up for user is enabled using the OTP process. If the e-mail is not existing, display an error message.
4	Event card (student home screen)	Click on the 'Event Card'.	Direct the user to the 'Registration Form' screen.
5	Submit form Button: 'Submit'	Fill form details. Click on the 'Submit' button.	Display 'successfully registered' if all the input fields are entered appropriately, otherwise display an error message.
6	Event card (Teacher home screen)	Click on the 'Event card'.	Direct the user to the 'Student List' screen.
7	Student button on the student list	Click on a student from the registered students list.	User directed to the screen where student registration details are displayed.
8	Button testing: Select or reject	Click on 'APPROVE' or 'DISAPPROVE'	Mail sent to the student whether selected or not. Message, 'APPROVED' or 'DISAPPROVED'. Remove registration details, of the student 'disapproved', from the database. Keep the student on the list if approved. Buttons disabled on clicking them once.

S/NO.	TEST	SAMPLE DATA	EXPECTED OUTCOME
9	Button testing: 'Add Event'	Click on 'Add Event'	User directed to the 'Add Event' screen.
10	Delete button	Click on the 'Delete' button that is on the event card on teacher's home screen.	Deletes the event and removes it from the home screen.
11	Button testing: 'Add Teacher'	Click on 'Add Teacher'	User directed to the 'Add Teacher' screen.
12	Button testing: 'View Teachers'	Click on 'View Teachers'	User directed to the 'View Teachers' screen.
13	Teacher button on the teacher list.	Click on a teacher from the view teachers list.	User directed to the 'Teacher Details' screen.
14	Button testing: 'Delete'	Click on the 'Delete' button.	Teacher removed from the list and teacher's data erased from the app and the database.
15	Button testing: 'View Admin'	Click on the 'View Admin' button and then on the admin.	User directed to the admin list. User directed to the 'View Admin' screen on clicking in the admin.
16	Button testing: 'Delete' and 'Add'	Click on the 'Delete' button. Then click on the 'Add' button.	User directed to the 'Add Admin' screen. Previous admin removed after adding a new admin successfully.
17	Log-out	Click on the 'Log-out' button.	Log-out the user and Direct to the 'Log-in' screen.

### **EXTENSIBILITY:**

My app will be flexible to extensions and changes, considering the designing and development process of the app. With the technologies and platforms, I have used, any changes or additions can be made to the app later if required by the client. Using PHP, the back-end becomes updatable; and using FLEX and react native, the front-end is available to changes, requiring any addition of screens or changes in design as per the client.