

Systems Report (Agile)

Topic: SRC Voting System

Class: 12SDD5

Group Number: 5

Group Members: Henil, Aryan

Client: Ms. Mercado

Supervisor: Mr. Montalban

A screenshot of a code editor showing a snippet of JavaScript code. The code is part of a file named 'index.html' and contains logic for handling drag-and-drop events on a file drop zone. It includes event listeners for 'dragenter', 'dragleave', 'dragover', and 'drop'. The 'dragover' event adds a 'solid-border' class to the drop zone. The 'drop' event removes this class and handles the dropped files using 'dataTransfer.files'. The 'dragleave' event removes the 'solid-border' class.

Auto-Table of Contents Page:

Project Information and Overview	2
1. Planning and Designing	4
1.1 - Needs Analysis (Requirements of user - Ms. Mercado)	4
1.2 - Algorithm - flowchart/pseudocode	4
1.3 - Variables and data types	4
1.4 - Interface Design	4
1.5 - Desk Check	5
1.6 - Context Diagram	5
1.7 - Data Flow Diagram	5
1.8 - Gantt Chart	5
2. Implementing	5
2.1 - Screenshots of program working	5
2.2 - Screenshots of code	6
3. Testing and Evaluating Software Solutions	6
3.1 - Evaluation of software against needs analysis	6

Stuff left and expected to change in sprint 2 (before week 3):

- Flowchart (bugs that we lost 1 mark in and also add the sessionStorage and localStorage part) - so create another mockup of flowchart 1 to get the idea behind this concept.
- What data type is radio button (section 1.3)
- Interface diagram (modify in terms of “review” votes) and edit the last one
- Add the idea of displaying “looks like currently, [MEMBER NAME] has the most 1st preferences and taking the lead”.
- Look at the DFD and see if the “localStorage” is needed to be added to the “storage” symbol
- ASK ABOUT GANTT CHART (do we create a new one for sprint 2)?
- Fix the bug of the email “jesusa.mercado@aphs.nsw.edu.au” CANNOT be stored in the localStorage.
- EXPLAIN the “evaluation” section explicitly (about 4 to 5 lines - 1 mark was lost here).
- The code: 1 mark was lost here (the last button didn’t work) - FIX THEM.
- Create new meeting minutes (for sprint 2). [DONE]

Project Information and Overview

- Project leader - Henil
- Software Development leader - Henil and Aryan
- Documentation leader - Aryan and Henil
- Division of labour:
 - Systems report sections:
 - Needs Analysis (Requirements) - H, A
 - Algorithms (Flowchart/pseudocode) - A
 - Context Diagram - A
 - Data-Flow Diagram - H
 - Variables and Data types - H, A
 - Interface Design - H, A
 - Desk check - H
 - Gantt chart - H, A
 - Implementation - Henil and Aryan
 - Evaluation of software against needs analysis - Henil and Aryan
 - Code - Henil
 - Communications with client - Henil and Aryan
- Group rules:
 - Respect each other
 - Listen actively to each other
 - If a problem occurs, then attack the problem, not the person
 - Communication and collaboration is key
 - Obey correct Workplace Health and Safety (WH&S) and Ergonomics procedures in the workplace

1. Planning and Designing

1.1 - Needs Analysis (Requirements of client - Ms. Mercado)

The purpose of this system is fully based on APHS. The aim of this system is to provide APHS with better software for students and staff to vote for SRC members every year (instead of voting on Google Forms). The needs/requirements of the system are given below in a table according to the client and based on client meetings.

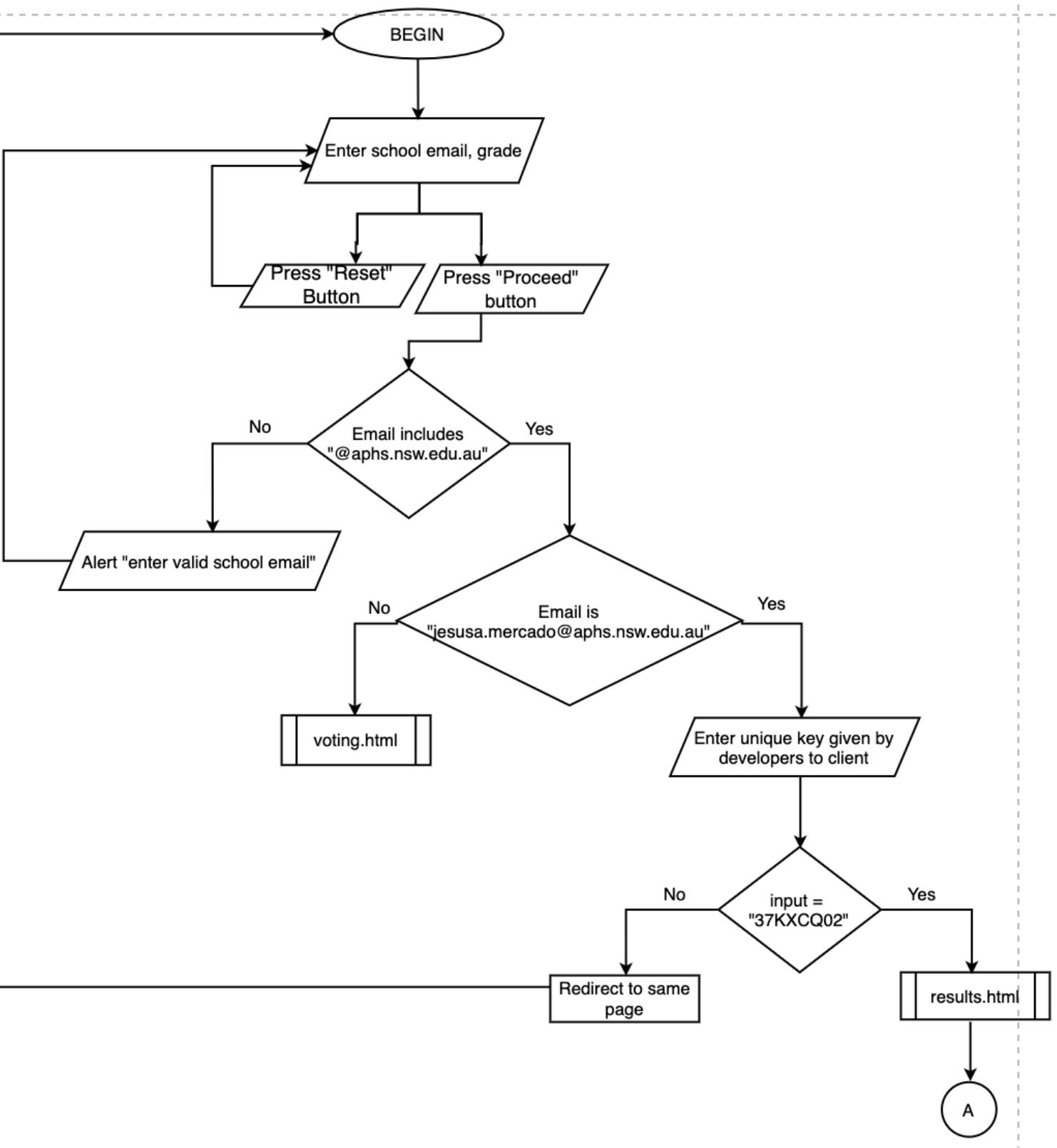
Must Have	Should Have	Could have
A preference and SRC members list that users can select.	A description and speech video of all the SRC members standing for vote.	A database that contains all the SRC members' details (like description, speech video, subjects they do, etc).
Only a student/staff of APHS can vote their preferences for the SRC members.	A prompt box appears for entering a unique key given to the client to ensure that the user is a client.	Member 1/7, 2/7, ... (keep up to time of what member is displayed each time).
The iteration must go on until a simple SRC member got the most effective preferences.		A tally system and results table that only gets displayed to Ms. Mercado and no one else.
Each member's container should appear one after other after the user selects his preference and then clicks the "next" button		

Please see both meeting minutes 1 and meeting minutes 2 document file in the folder to get a more depth and detailed view of needs/requirements of the client in regards to this system.

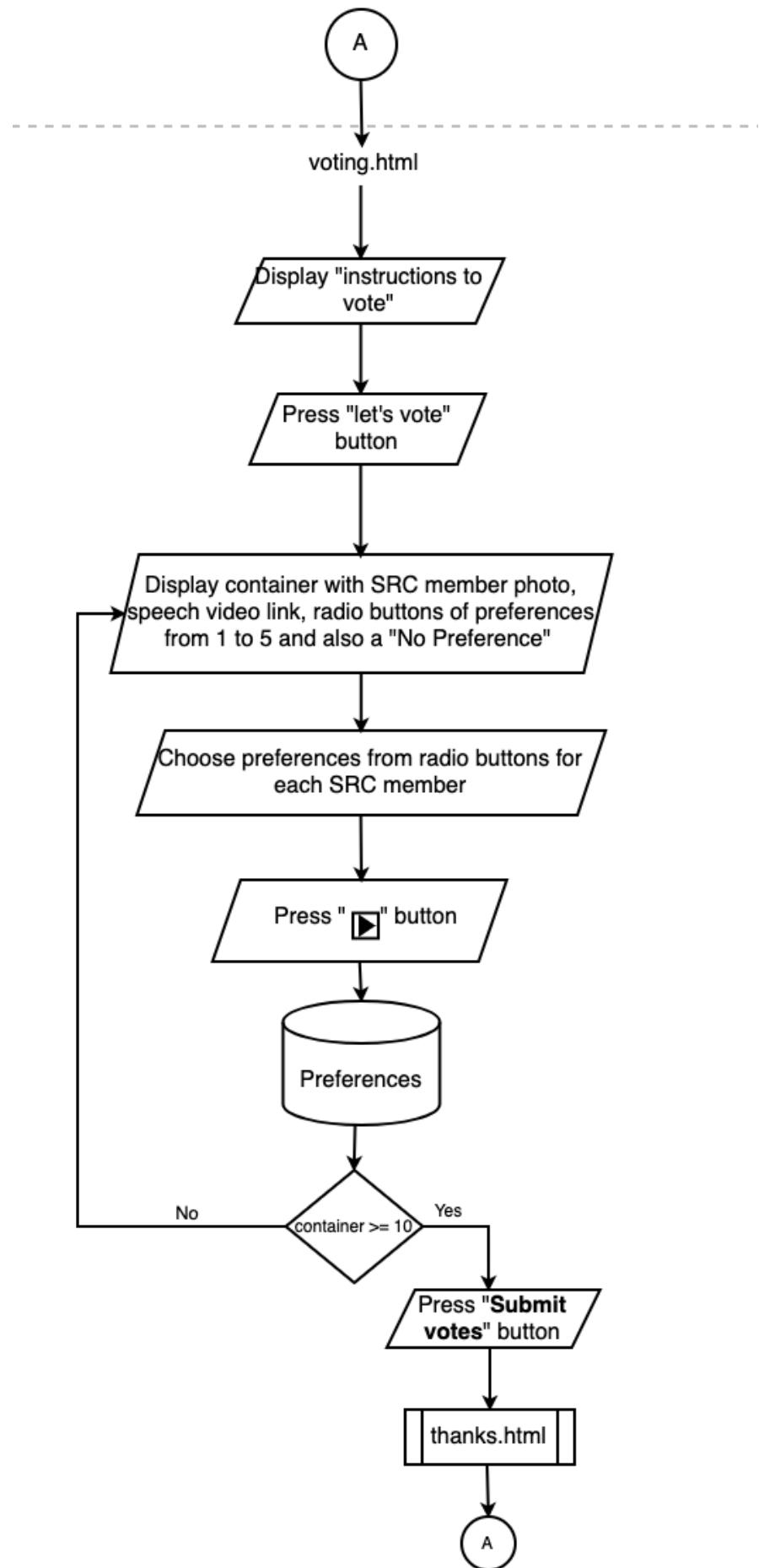
1.2 - Algorithm - flowchart/pseudocode

Link: [Our flowchart link \(on draw.io\)](#)

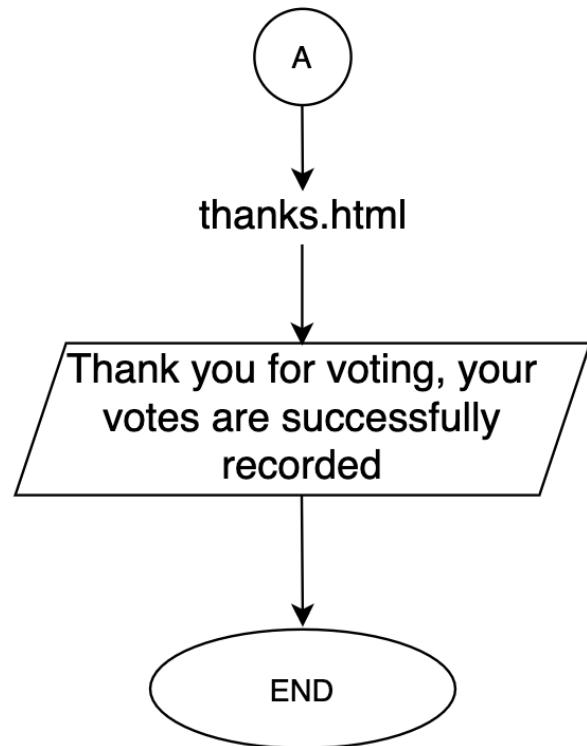
Flowchart 1: [*index.html*](#)



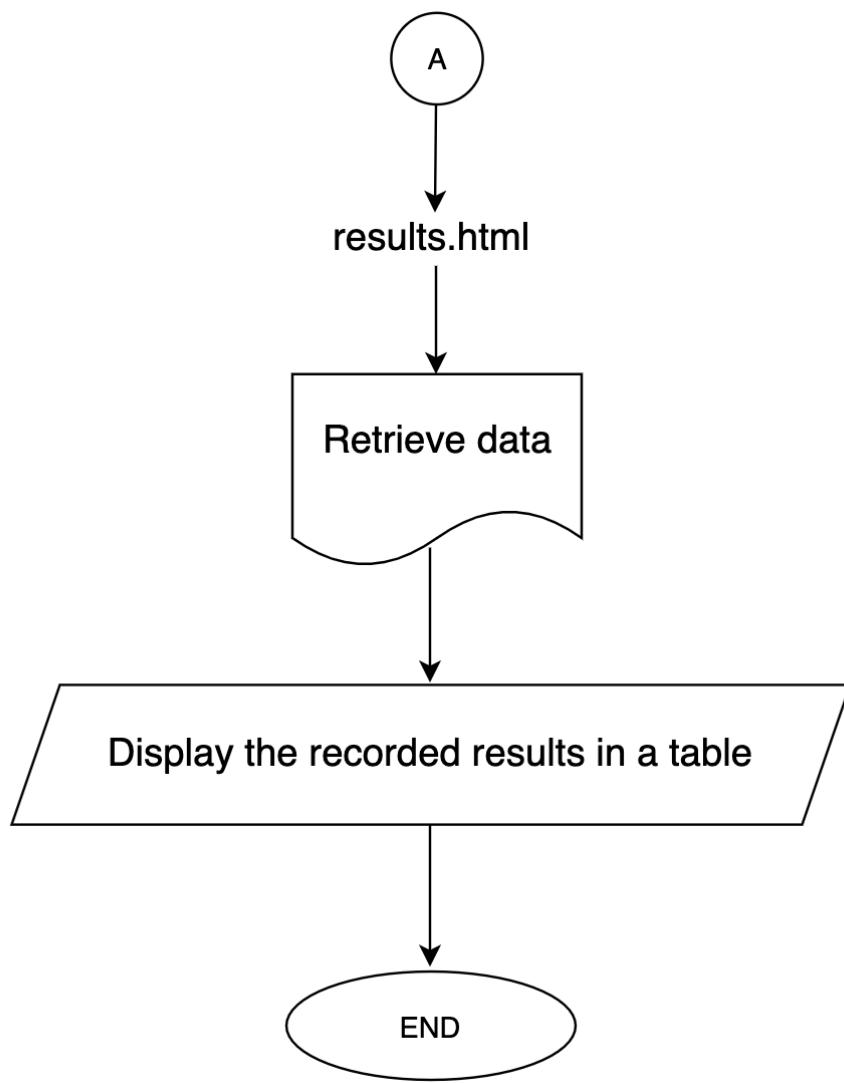
Flowchart 2: voting.html



Flowchart 3: thanks.html



Flowchart 4: results.html



1.3 - Variables and data types

Variable	Data Type	Description
mail	String	The email of the user who is voting their preferences on SRC members.
grade	Integer	The grade that the user is currently voting in APHS.
key	String	A unique identification key given only to the client to login and access the results.
preference	Enum/ boolean	The user preferences that basically identifies ranks of the SRC members they vote for.
details	String	

total	Integer	The final totals of each member's 1st, 2nd, 3rd, 4th, 5th and no preferences.
calculate	Integer	The calculations and processes of finding the SRC member who gets the highest 1st preferences in votes.
tally	String	The tallied display of the total of each member's 1st, 2nd, 3rd, 4th, 5th and no preferences.
result	Text	The final display of results of who gets the most 1st preferences and who is the captain so far until the client closes all the responses.

1.4 - Interface Diagram

Interface Diagram/Screen Design of index.html

Welcome to the Official School Representative Council Voting Website of
Arthur Phillip High School



Enter your email

Select your grade

← Animated background

Interface Diagram/Screen Design of voting.html

Instructions to vote:

1. Firstly, you can choose your preference.
2. Once you have chosen your preference for that SRC member, you can then click the "next member" button to continue voting and proceed to vote your preference for the next member.
3. Repeat these steps until you reach the last member preference list.

* NOTE: You can choose the "no preference" radio button as well if you think that you don't want to vote for a specific SRC member.

Interface Diagram/Screen Design of voting.html (continued) after pressing the "let's vote" button

Member 1 of 10

SRC Member 1

Picture of SRC Member 1

[MEMBER NAME]

[MEMBER DESCRIPTION]

[\[MEMBER SPEECH LINK\]](#)

Preferences

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>
No Preference	<input type="radio"/>

Previous Member 

Next Member 

Interface Diagram /Screen Design of thanks.html



Interface Diagram/Screen Diagram of results.html

User Email	User Grade	Preference for Charminkumar Patel	Preference for Henry Lim	Preference for Malika Azimi	Preference for Simardeep Gill	Preference for Mouna Matri	Preference for Lurcrecia Phung		
kraj@aphs.nsw.edu.au	11	1	2	4	4	3	2		
ecoco@aphs.nsw.edu.au	Teacher	2	3	5	No Preference	4	1		
tom.cooper@aphs.nsw.edu.au	10	3	2	3	4	No Preference	1		

1.5 - Desk Check

index.html								
Mail	Email includes == @aphs.nsw.edu.au	Email matches client's email	Enter unique pin given to client	Grade	Press "proceed" button	Proceed to:	Directed to next page	Display
alex.roy@gmail.com	False	False	N/A (Not a client email)	12	True	N/A	False	alert("Invalid email")
tcooper@aphs.nsw.edu.au	True	False	N/A (Not a client email)	11	True	voting.html	True	Takes the user to the next page
jesusa.mercado@aphs.nsw.edu.au	True	True	37KXCQ02	Teacher	N/A	results.html	N/A	N/A

voting.html

Display (Member Name)	Display (Member Description)	Display (Member photo)	Preference	Speech link	Press "next member" button	Container > 6	Press "Submit" votes button	Proceed to finish.html
PATEL, Charmin kumar	Studies in 12th grade, plays school cricket, does Modern history, SDD, IPT, Physics, Maths Ext 1...		1	True (For some reason, this member's speech link is not provided by the client to us)	True	False	False	False

LIM, Henry	Plays school Volleyball, can marry mathematics if he wants ...		2	True (Henry's speech link)	True	False	False	False
GILL, Simardeep	Likes to participate in extracurricular activities ...		No Preference	True (Simardeep's speech link)	True	False	False	False
AZIMI, Malika	Ambitious and passionate ...		4	True (Malika's speech link)	True	False	False	False
MAITRI, Mouna	Very passionate about working hard and the most preferable subject is English ...		5	False (Mouna's speech link)	True	False	False	False
PHUNG, Lucrecia	Interested in studying business studies ...		3	True (Lucrecia's speech link)	True	True	True	True

results.html

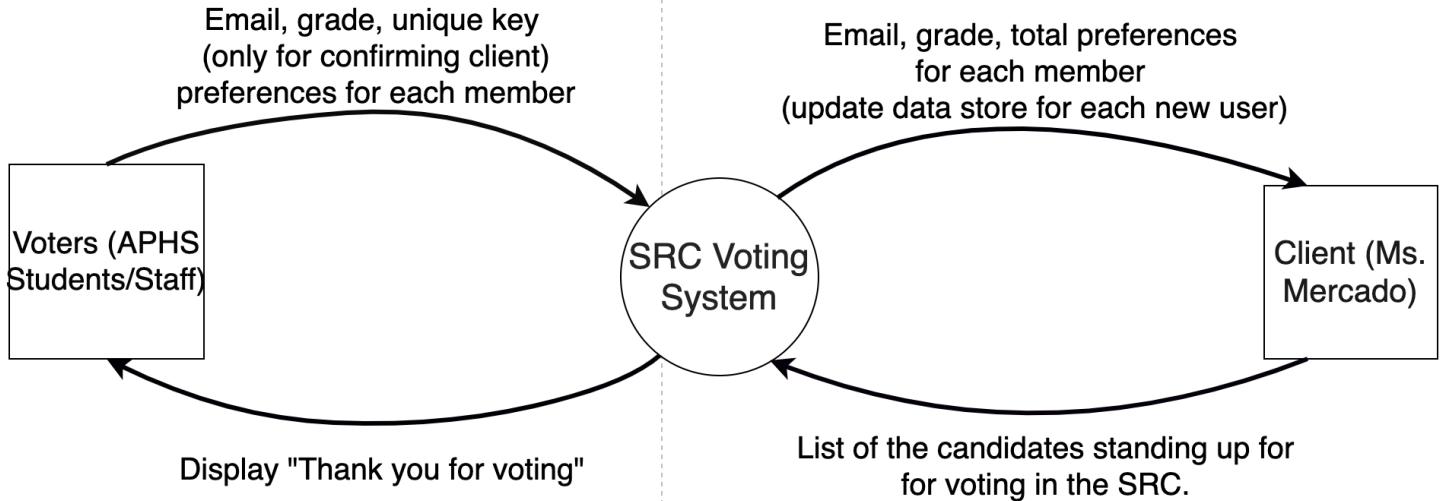
User email	User grade	Preference for Charmin	Preference for Henry	Preference for Malika	Preference for Simardeep	Preference for Mouna	Preference for Lucrecia
tom.cooper@aphs.nsw.edu.au	10	3	1	3	2	5	No preference
ecoco@aphs.nsw.edu.au	Teacher	2	1	1	2	3	4
pvraj@aphs.nsw.edu.au	11	2	3	No preference	1	4	5

finish.html

email != jesusa@mercado.aphs.nsw.edu.au	Finished voting preference for each SRC member	Display
TRUE	True	Thank you for voting, your preferences are successfully recorded.
FALSE	False	N/A
TRUE	False	N/A

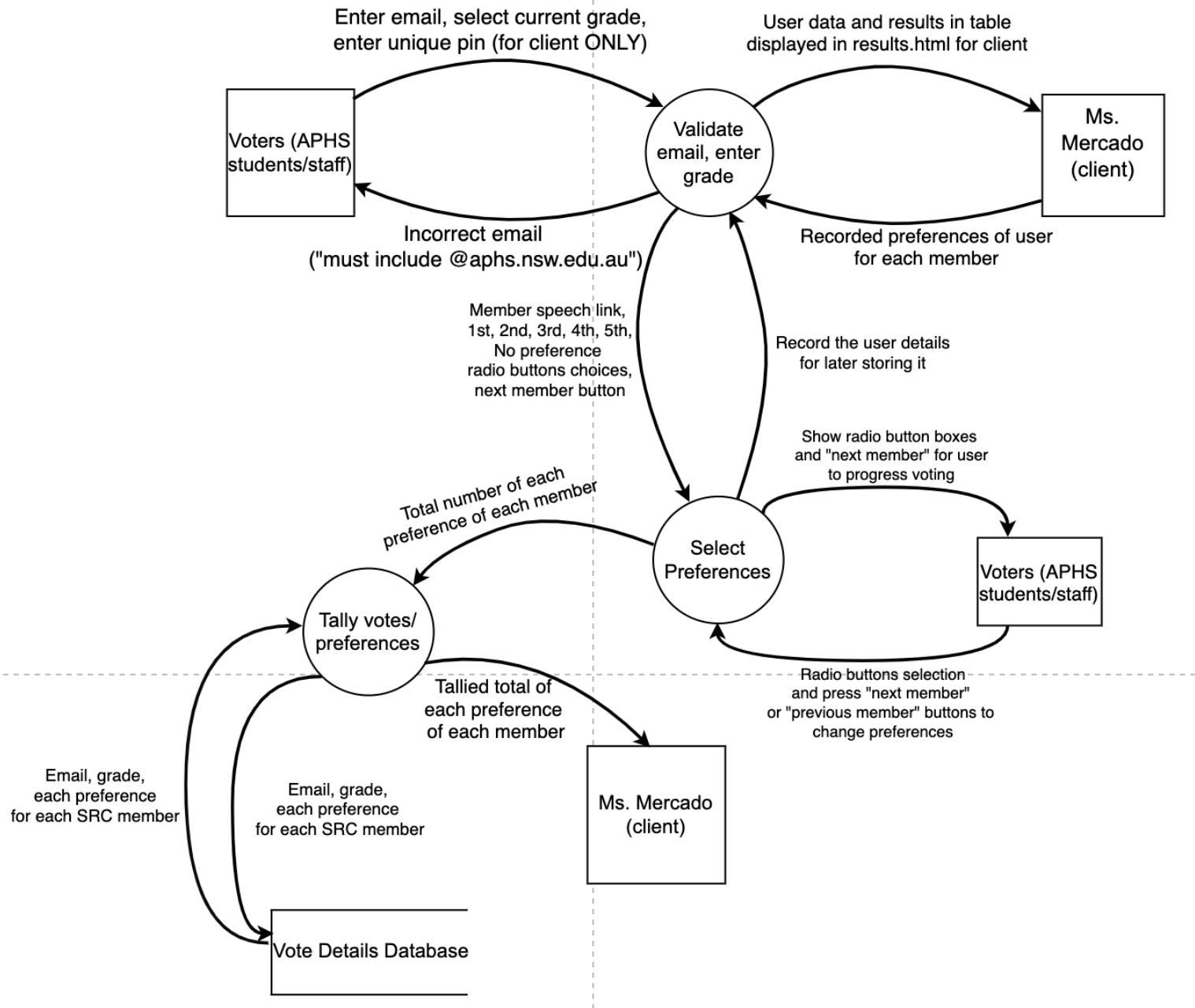
1.6 - Context Diagram

1.6 - Context Diagram for SRC Voting System



1.7 - Data Flow Diagram

1.7 - Data-Flow Diagram for SRC Voting System



1.8 - Gantt Chart

Topic: Developing Software Solutions							System Purpose					The purpose of this system is fully based on API better software for students and staff to vote for Google Forms.							
PROJECT TITLE		SRC Voting System					COMPANY NAME			APHS Tech									
PROJECT MANAGER		Henil and Aryan					DATE			25/02/21									
PHASE ONE																			
TASK NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION (Days)	TASK COMPLETION	Mon	Tue	Wed	Thur	Fri	Mon	Tue	Wed	Thur	Fri	Mon	Tue	Wed
1	Planning and Designing																		
1.1	Needs Analysis	Henil, Aryan	12/03/18	15/03/18	3	100%													
1.2	Algorithm - flowchart/pseudocode	Aryan	15/03/18	16/03/18	1	100%													
1.3	Variables and data types	Henil	15/03/18	21/03/18	6	90%													
1.4	Interface Design	Henil, Aryan	16/03/18	22/03/18	6	40%													
1.5	Desk Check	Henil	18/03/18	22/03/18	4	70%													
1.6	Context Diagram	Aryan	19/03/18	22/03/18	3	60%													
1.7	Data Flow Diagram	Aryan			0														
1.8	Gantt Chart	Henil, Aryan	23/03/18	23/03/18	0	50%													
2	Implementing																		
2.1	Screenshots of program working	Henil	24/03/18	28/03/18	4	22%													
2.2	Screenshots of code	Henil	29/03/18	02/04/18	3	16%													
3	Testing and Evaluating																		
3.1	Evaluation of Software against Needs Analysis	Henil, Aryan			0	0%													

Link to access the Gantt Chart

⇒ [Our Gantt Chart link](#)

2. *Implementing*

2.1 - Screenshots of program working

1) Index.html (beginning page of system)

Screenshot 1: beginning page

Welcome to the official School Representative Council Voting website of

Arthur Phillip High School



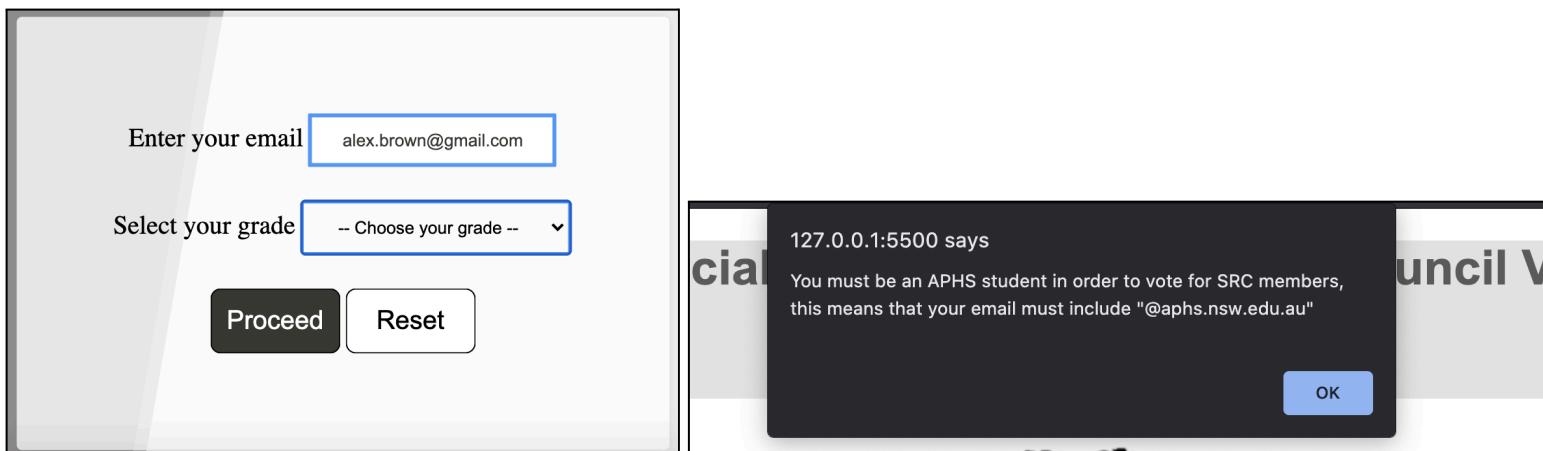
Enter your email

Select your grade

Explanation of screenshot 1 -

This is the title page of the SRC voting system. This page has a title and an 'Arthur Philip High School' logo. This page is used as the first page for user login and verification. The main reason is that the client can easily access information by entering her email on this page. The user first enters his/her email address and then selects their grade and presses the submit button. The reason for choosing such a live background with a neutral gray and a cool white colour to become an attraction to the user.

Screenshot 2: User email input validation



Explanation of screenshot 2 -

The image above shows how the user has inputted their email address and left the grade blank, this is because the system itself sets the grade to 'Grade 7' which is a bug, the grade if not entered should also have a alert box showing that 'The grade was not entered'. There are two buttons under the 'Select your grade option' where the both the buttons have a black and white hovering effect. The proceed button guides the user to the next page of the system, whereas the reset button will reset the email and grade to a blank space. As we saw in the image beside the email entered was 'alex.brown@gmail.com' whereas the email should contain '@aphs.nsw.edu.au'. The reason for alerting the user is that the user who is voting must be a member of our school as per the clients requirement, so that a user outside APHS cannot access and vote for the SRC members.

Screenshot 4: Inputting user details (email, grade)

The image consists of two side-by-side screenshots of a web application interface. Both screenshots show a form with fields for 'Enter your email' and 'Select your grade'. In the first screenshot, the 'Enter your email' field contains 'example@aphs.nsw.edu'. A dropdown menu for 'Select your grade' is open, showing options: '7', '8', '9', '10', '11', '12', and 'Teacher'. The 'Proceed' button is visible below the dropdown. In the second screenshot, the 'Enter your email' field now contains 'rcado@aphs.nsw.edu.au'. The dropdown menu is closed, showing 'Teacher' as the selected option. Below the dropdown are 'Proceed' and 'Reset' buttons.

Explanation of screenshot 4 -

The above images show that the client is about to request the system to access the voting system results. She enters her email address and adds as a 'Teacher' in the 'Enter your grade' field. Then she presses the proceed button. The reason for doing so was that it made it easier for the client to access the results instead of going through the whole voting process and then accessing the results.

Screenshot 5: Inputting unique key:

The image consists of two side-by-side screenshots of a web application interface. Both screenshots show a modal dialog box with the text '127.0.0.1:5500 says' at the top and 'Enter unique key' below it. In the first screenshot, there is an empty text input field. In the second screenshot, the text input field contains the value '37KXCQ02'. At the bottom of each dialog are 'Cancel' and 'OK' buttons.

Explanation of screenshot 5 -

After the client enters her email address and the system checks her email address so that it is a valid email. The images shown are alert boxes which appear as soon as the teacher puts her email address in 'Enter you email', then the system will ask the client to 'Enter a unique key' this will help the system to cross check that it is actually the client who is accessing the results. The reason for adding this security was to protect the results as anyone other than the client can enter the client's email and access the results which would make the results public.

2) voting.html (second page of system)

Screenshot 1: Displaying instructions of how to vote

Instructions for voting:

1. Firstly, you can choose your preference according to the SRC member
2. Once you have chosen your preference for that SRC, you can then click "next member" button
3. You will then be directed to the next member preference list and then you may proceed to choose your preference
4. Repeat these steps until you reach the last member preference list

NOTE: You can choose the "no preference" radio button as well if you think that you don't want to vote for a specific SRC member

Let's vote

Explanation of screenshot 1 -

The image above shows that after the user inputs his/her details and they are validated the user is redirected to the next page that is this page. The reason for making this page before directly displaying the candidates. The user reads all the information for the voting process, and then has to press the 'Lets Vote' button. The button and all the instructions disappear and displays the final process of voting for the candidates.

Screenshot 2: SRC Member Preference container

SRC Member 1



PATEL, Charminkumar

Charminkumar Patel studies in 12th grade and is diligent and tries his best to achieve well in all his subjects and HSC.

[Member's Speech link](#)



Preference:

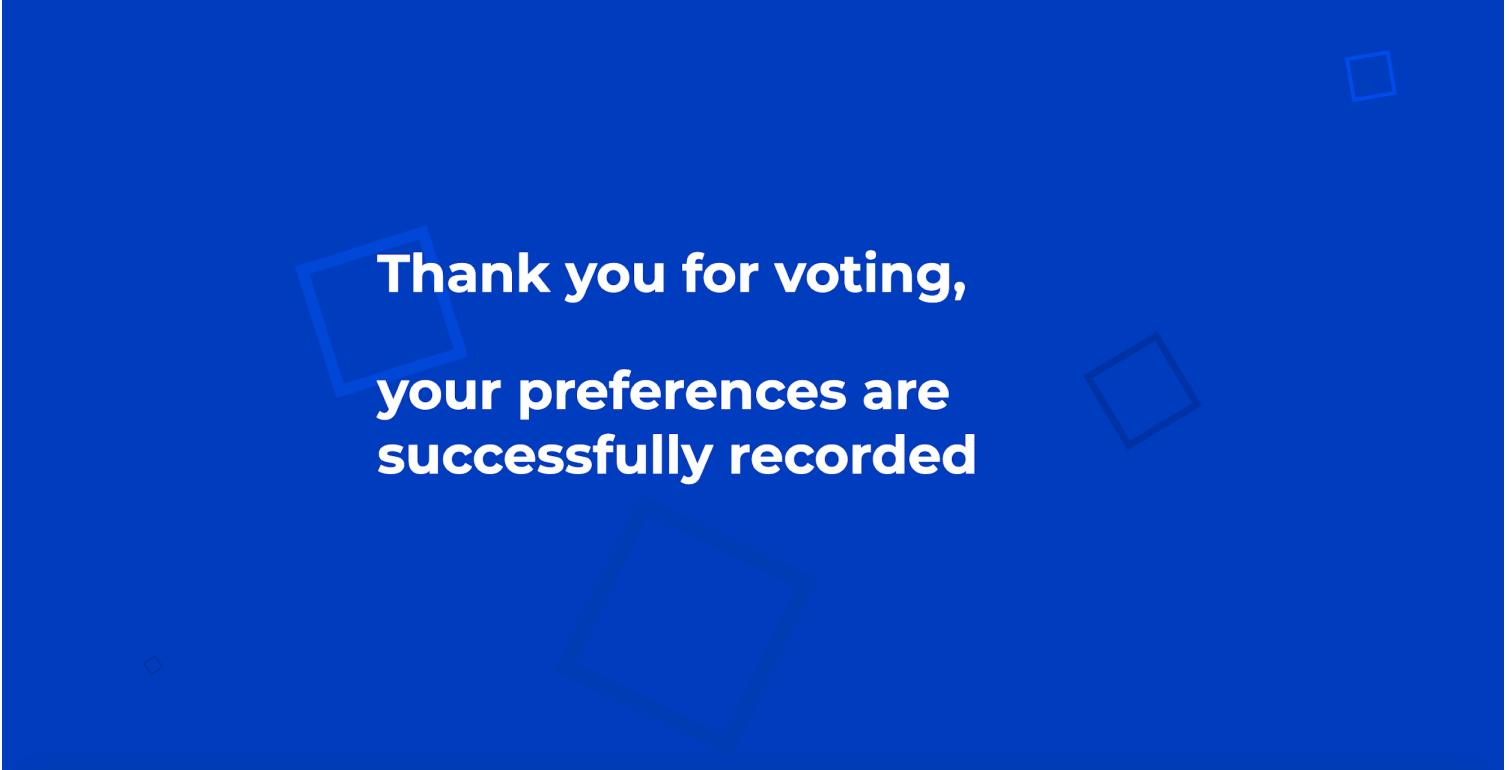
1	<input checked="" type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>
No Preference	<input type="radio"/>

Explanation of screenshot 2 -

This page is the main page in the SRC voting system, users have to give preferences to each candidate and if they do not wish to provide a given candidate a preference then he/she can click on 'No Preference'. The reason for giving almost all the information to the user is that he/she can make a better decision in choosing their leader. The user then has to press the arrow button to proceed to the next member and give a preference for the next candidate. The user is provided with a photo of the candidate, his full name and a short description of him/her. Users are also provided with the candidates speech link,

which then redirects the user to the speech link, where they can listen to the whole speech, this idea was chosen to better help the user to listen to their speech if he/she has missed listening to their speech.

3) thanks.html (third page of the SRC Voting System)



**Thank you for voting,
your preferences are
successfully recorded**

Explanation of above screenshot -

The above image shows the final page for the system. The system displays 'Thank you for voting" to the user's screen. This page doesn't work currently as this is just a prototype and the production and development of this will be continued in the next sprint (sprint 2).

4) results.html (only Ms. Mercado can access this page)

2.2 - Screenshots of code

Style.css for index page page (background animation and CSS):

```
<style>

/* Filename: style.css */

html {
    margin: 0;
    padding: 0;
}

/* body {
    margin: 0;
} */



#countdown_Timer {
    position: absolute;
    text-align: center;
    font-size: 55px;
    margin: 0 auto;
    display: inline-block;
}

/* Animation CSS code for animating background */
.bg {
    animation: slide 4.5s ease-in-out infinite alternate;
    background-image: linear-gradient(-80deg, white 50%, grey 50%);
    bottom: 0;
    left: -50%;
    opacity: .5;
    position: fixed;
    right: -50%;
    top: 0;
    z-index: -1;
}

.bg2 {
    animation-direction: alternate-reverse;
    animation-duration: 5.5s;
}

.bg3 {
    animation-duration: 6.5s;
}

.content {
    background-color: rgba(255, 255, 255, .8);
    border-radius: .25em;
```

```
box-shadow: 0 0 .50em rgba(136, 113, 113, 0.199);
box-sizing: border-box;
left: 50%;
padding: 10vmin;
position: fixed;
text-align: center;
top: 50%;
transform: translate(-50%, -10%);
}

h1 {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 35px;
}

img {
    height: auto;
    width: auto;
}

#mail {
    border: 3px solid green;
    border-radius: 5px;
    padding: 10px 22px;
    text-align: center;
    margin-bottom: 25px;
    width: auto;
    background-color: rgb(58, 54, 54);
    color: rgba(255, 255, 255, 0.938);
}

.user_mail {
    color: black;
    font-size: 20px;
    border-radius: 12px;
}

#grade {
    border: 3px solid green;
    border-radius: 3px;
    padding: 10px 22px;
    text-align: center;
    margin-bottom: 25px;
    background-color: transparent;
    color: black;
}
```

```

.voter_grade {
    color: black;
    font-size: 20px;
    border-radius: 12px;
}

@keyframes slide {
    0% {
        transform: translateX(-25%);
    }

    100% {
        transform: translateX(25%);
    }
}

```

Explanation of above code snippet:

The above is a .css file and the code in it is adding styles to the background of the index.html page (first login page for users). The background is animated.

Index.html (standard login page for users):

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script language="javascript" type="text/javascript" src="page1.js"></script>
    <script language="javascript" type="text/javascript" src="vote.js"></script>
    <link rel="stylesheet" type="text/css" href="style.css">

    <link rel="shortcut icon" type="image/png"
    href="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR83KLactzDffcyV-DAKcgxilXui9i8bm3mrQ&us
    qp=CAU">
    <title>SRC Voting System</title>
</head>

<body>
    <!-- Greeting to user -->
    <h1>Welcome to the official School Representative Council Voting website of <br><p
    id="aphs">Arthur Phillip High School</p></h1>

```

```

<!-- APHS School logo -->

<!-- For background animation purposes -->
<div class="bg"></div>
<div class="bg bg2"></div>
<div class="bg bg3"></div>
<div class="content"> <!-- A small container in which the form input fields and buttons will get displayed -->

<form>
    <!-- Input for getting user email -->
    <div>
        <label for="mail" class="user_mail">Enter your email</label>
        <input id="mail" name="mail" type="email" placeholder="example.vote@aphs.nsw.edu.au" autofocus required>
    </div>

    <!-- Drop-down box for selection of user's grade -->
    <div>
        <label for="grade" class="voter_grade">Select your grade</label>
        <select name="grade" id="grade">
            <option value="" selected disabled>-- Choose your grade --</option> <!-- Default value of drop-down box -->
            <option value="7" label="7"></option><br>
            <option value="8" label="8"></option><br>
            <option value="9" label="9"></option><br>
            <option value="10" label="10"></option><br>
            <option value="11" label="11"></option><br>
            <option value="12" label="12"></option><br>
            <option value="Teacher" label="Teacher"></option><br> <!-- Even teacher/staff can vote for SRC members -->
        </select>
    </div>

    <!-- For security purposes, a unique key to confirm that client's email is not unauthorisedly accessed -->
    <div id="unique_key">
        <label for="unique" class="uniquekey">Enter the unique identification key given to you</label>
        <input id="unique" name="unique" type="text" placeholder="1-EX-AM-PLE-07">
    </div>

```

```

<div>
    <!-- "proceed" button to go to next page according to validation and "reset" button
to reset the details inputted -->
    <button type="submit" id="submit" onclick="next()">Proceed</button>
    <button type="reset" id="reset" onclick="restart()">Reset</button>
</div>

</form>
</div>

```

Explanation of above code snippet (index.html)

The above code snippet is creating a HTML page and context of input fields for user email, and a drop-down box for users selecting their grades (also Teachers are allowed to vote).

Page1.js (all the validations including email, grade, unique key)

```

// Filename: page1.js //
function next() {
    email = document.getElementById('mail').value;

    // Nested if statements //
    if (email == "jesusa.mercado@aphs.nsw.edu.au") {
        // For client security //
        id = "37KXCQ02"; // Given only to client to access results //

        var unique = prompt("Enter unique key"); // User input of given unique key to client to
confirm it is the client //

        if (unique == id) {
            window.open("results.html"); // Direct the user (client) to results.html page //
        }

        else {
            location.href = "#"; // Reload user's page if he/she doesn't enter the correct pin
as per client security purposes //
        }
    }

    else {
        // For school security //
        if (email.includes("@aphs.nsw.edu.au")) {

```

```

        window.open("./voting.html"); // Direct the user to voting.html if his/her email
includes @aphs.nsw.edu.au //
    }

    else {
        alert("You must be an APHS student in order to vote for SRC members, this means that
your email must include \"@aphs.nsw.edu.au\"");
        location.href = "#"; // Reload user's page and don't let him/her go to the next page
if email doesn't include @aphs.nsw.edu.au //
    }
}
}
}

```

Explanation of above code snippet (page1.js):

The above javascript code is a very important part of this system as it validates the email input field and secures an unauthorised access in order to vote. It states that if a random user enters client's email (Ms. Mercado's email), then he/she will have to enter a unique key that is given to the client only.

voting.html (instructions for user to vote):

```

<div id="instr">
    <ol>
        <li>Firstly, you can choose your preference according to the SRC member</li><br>
        <li>Once you have chosen your preference for that SRC, you can then click "next member"
button</li><br>
        <li>You will then be directed to the next member preference list and then you may
proceed to choose your preference</li><br>
        <li>Repeat these steps until you reach the last member preference list</li><br>
        <p>NOTE: You can choose the "no preference" radio button as well if you think that you
don't want to vote for a specific SRC member</p><br>
    </ol>
</div>

```

Explanation of above code snippet:

The above code basically displays the instructions for the users to vote and steps.

voting.html (container for each SRC Member preferences and details):

```
<div id="container_1">
    <h2>SRC Member 2</h2>
    
    <p>AZIMI, Melika</p>
    <p>Compassionate and optimistic about her future, no-one can let her down, not even herself.</p>
    <a href="https://drive.google.com/file/d/1o1pGcYSJPLt7WB1KnP0R9yxA-H3fCJou/view">Member's Speech Link</a>

    <div class="form">
        Preference:
        <div class="inputGroup">
            <input id="one1" name="radio_one" type="radio"/>
            <label for="one1">1</label>
        </div>

        <div class="inputGroup">
            <input id="two1" name="radio_one" type="radio"/>
            <label for="two1">2</label>
        </div>

        <div class="inputGroup">
            <input id="three1" name="radio_one" type="radio"/>
            <label for="three1">3</label>
        </div>

        <div class="inputGroup">
            <input id="four1" name="radio_one" type="radio"/>
            <label for="four1">4</label>
        </div>

        <div class="inputGroup">
            <input id="five1" name="radio_one" type="radio"/>
            <label for="five1">5</label>
        </div>

        <div class="inputGroup">
            <input id="NA2" name="radio_one" type="radio"/>
            <label for="NA2">No Preference</label>
        </div>
    </div>
</div>
```

Explanation of above code snippet:

The above code basically generates after the “let’s vote” button is pressed after reading the instructions of how to vote. This code runs in a div container where SRC member’s details like (member name, description, SRC photo, SRC speech video link) so that users can get clear information about the members. Radio buttons are displayed to users to select one preference for the SRC member.

voting.html (for pressing “previous member” button and “next member” button):

```
<div id="member-2">
    <h2 id="back">Previous Member</h2>
    <button id="member_two" onclick="member_2()"><i class="far fa-caret-square-left" id="icon_2"></i></button>
</div>

<div id="member_2">
    <h2 id="proceed">Next Member</h2>
    <button id="member2" onclick="member2()"><i class="far fa-caret-square-right" id="icon2"></i></button>
</div>
```

Explanation of above code snippet:

The above code defines that if the “previous member” button is pressed, then the user will go back to the previous member. Also “next member” button is pressed, then the user will go back to the next member.

style.css (code for radio buttons design and selections):

```
.inputGroup {
    background-color: #fff;
    display: block;
    margin: 10px 0;
    position: relative;
}

.inputGroup label {
    padding: 12px 20px;
    width: 100%;
    display: block;
    text-align: left;
    color: #3C454C;
    cursor: pointer;
    position: relative;
```

```

    z-index: 2;
    transition: color 200ms ease-in;
    overflow: hidden;
}

.inputGroup label:before {
    width: 10px;
    height: 10px;
    border-radius: 50%;
    content: "";
    background-color: #5562eb;
    position: absolute;
    left: 50%;
    top: 50%;
    transform: translate(-50%, -50%) scale3d(1, 1, 1);
    transition: all 300ms cubic-bezier(0.4, 0, 0.2, 1);
    opacity: 0;
    z-index: -1;
    /* border: 2px solid black;
    border-radius: 8px; */
}

.inputGroup label:after {
    width: 32px;
    height: 32px;
    content: "";
    border: 2px solid #D1D7DC;
    background-color: #fff;
    background-image: url("data:image/svg+xml,%3Csvg width='32' height='32' viewBox='0 0 32 32' xmlns='http://www.w3.org/2000/svg'%3E%3Cpath d='M5.414 11L4 12.414L5.414 5.414L20.828 6.414 19.414 51-10 10z' fill='%23fff' fill-rule='nonzero'/%3E%3C/svg%3E ");
    background-repeat: no-repeat;
    background-position: 2px 3px;
    border-radius: 50%;
    z-index: 2;
    position: absolute;
    right: 30px;
    top: 50%;
    transform: translateY(-50%);
    cursor: pointer;
    transition: all 200ms ease-in;
}

.inputGroup input:checked ~ label {
    color: #fff;
}

```

```
.inputGroup input:checked ~ label:before {
    transform: translate(-50%, -50%) scale3d(56, 56, 1);
    opacity: 1;
}

.inputGroup input:checked ~ label:after {
    background-color: #54E0C7;
    border-color: #54E0C7;
}

.inputGroup input {
    width: 32px;
    height: 32px;
    order: 1;
    z-index: 2;
    position: absolute;
    right: 30px;
    top: 50%;
    transform: translateY(-50%);
    cursor: pointer;
    visibility: hidden;
}

.form {
    padding: 0 16px;
    max-width: 550px;
    margin: 50px auto;
    font-size: 18px;
    font-weight: 600;
    line-height: 36px;
}

body {
    background-color: #D1D7DC;
    font-family: "Fira Sans", sans-serif;
}

 *,
 *::before,
 *::after {
    box-sizing: inherit;
}

html {
    box-sizing: border-box;
```

```
}

code {
    background-color: #9AA3AC;
    padding: 0 8px;
}
```

Explanation of above code snippet:

The above code styles the radio buttons and the selection effects.

style.css (for not displaying all the divs at once):

```
#container {
    display: none;
}

#container_1 {
    display: none;
}

#container_2 {
    display: none;
}

#container_3 {
    display: none;
}

#container_4 {
    display: none;
}

#container_5 {
    display: none;
}
```

Explanation of above code snippet:

Initially, all the containers are displayed as invisible and are related to the event listeners of buttons.

vote.js (for adding functionality to “next member” button):

```
// After pressing the "let's vote" button as an event listener //
function move() {
    document.getElementById("instr").style.display = "none"; // disappear instructions //
    document.getElementById("container").style.display = "block"; // display first container //
    document.getElementById("member1").style.display = "block"; // display "next member" button as
id of "member1"
    document.getElementById('move').style.display = "none"; // disappear the "let's vote" button
    document.getElementById('icon1').style.display = "block"; // display the right-sided arrow from
HTML //
    document.getElementById('icon_1').style.display = "block"; // display the left-sided arrow from
HTML //
}

// After pressing the "next member" button as an event listener //
function member1() {
    document.getElementById('container').style.display = "none";
    document.getElementById('container_1').style.display = "block";
    document.getElementById('member1').style.display = "none";
    document.getElementById('member2').style.display = "block";
    document.getElementById('icon2').style.display = "block";
    document.getElementById('icon_2').style.display = "block";
}

function member2() {
    document.getElementById('member2').style.display = "none";
    document.getElementById('container_1').style.display = "none";
    document.getElementById('container_2').style.display = "block";
    document.getElementById('member3').style.display = "block";
    document.getElementById('icon3').style.display = "block";
    document.getElementById('icon_3').style.display = "block";
}

function member3() {
    document.getElementById('member3').style.display = "none";
    document.getElementById('container_2').style.display = "none";
    document.getElementById('container_3').style.display = "block";
    document.getElementById('member4').style.display = "block";
    document.getElementById('icon4').style.display = "block";
    document.getElementById('icon_4').style.display = "block";
}

function member4() {
    document.getElementById('member4').style.display = "none";
    document.getElementById('container_3').style.display = "none";
    document.getElementById('container_4').style.display = "block";
```

```

document.getElementById('member5').style.display = "block";
document.getElementById('icon5').style.display = "block";
document.getElementById('icon_5').style.display = "block";
}

function done() {
    location.href = "thanks.html"; // take the user to thanks.html page //
}

```

Explanation of above code snippet:

These are event handlers/listeners that function for the code to run logically.

thanks.html (user finishes voting):

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script language="javascript" type="text/javascript" src="vote.js"></script>

    <title>Thank you</title>
</head>
<body>
    <h1 id="thank"></h1>

    <div class="hero">
        <div class="hero__title">Thank you for voting, <br><p>your preferences are successfully recorded</p></div>
        <div class="cube"></div>
        <div class="cube"></div>
        <div class="cube"></div>
        <div class="cube"></div>
        <div class="cube"></div>
        <div class="cube"></div>
    </div>

    <!-- The idea is to redirect user back to the first page but he/she CANNOT vote again and this is checked by the Javascript whether the user has entered before and voted, so matching the user's email and grades is important for Javascript -->

```

This code displays thank you to the user after the user completes the whole process for voting.

CSS in thanks.html (animated background):

```
<style>

@import url("https://fonts.googleapis.com/css?family=Montserrat:700");

.hero {
    background-color: #0040C1;
    position: relative;
    height: 100vh;
    overflow: hidden;
    font-family: "Montserrat", sans-serif;
}

.hero__title {
    color: #fff;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    font-size: 50px;
    z-index: 1;
}

.cube {
    position: absolute;
    top: 80vh;
    left: 45vw;
    width: 10px;
    height: 10px;
    border: solid 1px #003298;
    transform-origin: top left;
    transform: scale(0) rotate(0deg) translate(-50%, -50%);
    -webkit-animation: cube 12s ease-in forwards infinite;
        animation: cube 12s ease-in forwards infinite;
}

.cube:nth-child(2n) {
    border-color: #0051f4;
}

.cube:nth-child(2) {
    -webkit-animation-delay: 2s;
        animation-delay: 2s;
    left: 25vw;
    top: 40vh;
}

.cube:nth-child(3) {
    -webkit-animation-delay: 4s;
        animation-delay: 4s;
}
```

```

        animation-delay: 4s;
left: 75vw;
top: 50vh;
}
.cube:nth-child(4) {
-webkit-animation-delay: 6s;
    animation-delay: 6s;
left: 90vw;
top: 10vh;
}
.cube:nth-child(5) {
-webkit-animation-delay: 8s;
    animation-delay: 8s;
left: 10vw;
top: 85vh;
}
.cube:nth-child(6) {
-webkit-animation-delay: 10s;
    animation-delay: 10s;
left: 50vw;
top: 10vh;
}

@-webkit-keyframes cube {
from {
    transform: scale(0) rotate(0deg) translate(-50%, -50%);
    opacity: 1;
}
to {
    transform: scale(20) rotate(960deg) translate(-50%, -50%);
    opacity: 0;
}
}

@keyframes cube {
from {
    transform: scale(0) rotate(0deg) translate(-50%, -50%);
    opacity: 1;
}
to {
    transform: scale(20) rotate(960deg) translate(-50%, -50%);
    opacity: 0;
}
}

/* body {

```

```

background-image:
url('https://flevix.com/wp-content/uploads/2019/12/bubble-animated-background.svg');
} */

#thank {
    text-align: center;
    position: absolute;
    transform: translateY(50%);
}
</style>
```

This code is mainly used for designing the background of the page to make it more attractive.

results.html (uncompleted - will get continued in sprint 2):

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>This are the results for the preferences of each of the SRC member</title>
</head>

<body id="results">
    <h1>Results by Tally (only the client Ms. Mercado can see this page)</h1>
    <div w3-include-html="voting.html"></div>

    <p id="hidden"></p>

    <p id="one"></p>
```

3. Testing and Evaluating Software Solutions

3.1 - Evaluation of software against needs analysis

- Did the prototype of the system meet Ms. Mercado's initial requirements?

Number	Met or Not Met	Requirements and needs of Software
1	Met	The email validates which means that it only accepts emails with “@aphs.nsw.edu.au”
2	Not Met	The grade drop-down box validates well so that users will have to select that grade other than the default which is “-- Select your grade --”
3	Met	The reset button and proceed button function as expected on the first page according to the event handlers.
4	Met	The client's email that is ' jesusa.mercado@aphs.nsw.edu.au ' validates and then asks for a pin to confirm that the client's email is not accessed unauthorisedly. NOTE: The unique pin is given to the client by the developers.
5	Not Met yet	The user data also gets stored (email, grade, preferences for each member) in the page1.JS file.
6	Met	SRC Members' Picture and their Speech Link get displayed in each member's container in their appropriate place.
7	Not Met	The data of the user's preferences from radio buttons get stored in the vote.js file and retrieved in the results.html file.
8	Met	The position of the 'Next Member' and 'Previous Page' are in their appropriate place where the “previous member” arrow must be sharp on the left and vice versa.
9	Met	The instructions disappear as soon as the user presses “ Let's vote ”.
10	Not Met	The position of the 'Next Member' and 'Previous Page' are in the center of the page.

Therefore, the requirements that are not met clearly are still gonna get carried on to sprint 2 of this project next term and will be completed and met successfully.

