Emergency Class Manager: Requirements Document

By: Katie Hoskins, Tanya Peacock, Marcus Strange

Table of Contents

Project Name

Our website is for schools or school districts that want to use a simple website instead of a basic spreadsheet to keep track of students in emergencies. It allows for teachers and administrators to manage their class which is why we came up with the name **Emergency Class Manager**.

Team Member Names

See **Figure 1** in the appendix for a detailed description.

Abstract

The Emergency Class Manager project will allow teachers to have a centralized space to keep track of their students in emergency situations. Teachers will have the ability to access the website on mobile or on a computer. They will have a place to check off students as present, missing, or absent and then submit that information to the administration. This will be a more streamlined process for teachers as well as administrators to keep track of students. The reports will be sent to the administrator at the end of the drill or emergency. When the need arises, this will be a much simpler and more effective way of accounting for students.

Tools and Technologies

See **Figure 2** in the appendix for a detailed description.

Requirements List

1. Registration Form

1.1. The registration form will be one component with the following:

1.1.1. A header text of “Sign Up”

1.1.2. A label for First Name

1.1.3. A box for First Name

1.1.3.1. The box will accept a first name of at least two letters

1.1.3.1.1. On invalid input, the user will be notified that there must be at least two letters in the first name

1.1.3.2. The box will only accept characters

1.1.3.2.1. On invalid input, the user will be notified that there may only be characters used in the first name

1.1.3.3. The box will be marked as required with an *

1.1.4. A label for Last Name

1.1.5. A box for Last Name

1.1.5.1. The box will accept a last name of at least two letters

1.1.5.1.1. On invalid input, the user will be notified that there must be at least two letters in the last name

1.1.5.2. The box will only accept characters

1.1.5.2.1. On invalid input, the user will be notified that there may only be characters used in the last name

1.1.5.3. The box will be marked as required using *

1.1.6. A label for an email address

1.1.7. An input box for an email address

1.1.7.1. The box will accept a valid email address through the school system

1.1.7.1.1. Emails must contain an @ symbol

1.1.7.1.2. The email must end in the domain cmcss.net

1.1.7.1.3. The email will be verified by the server

1.1.7.2. The box will be marked as required using *

1.1.8. A label for Password

1.1.9. An input box for Password

1.1.9.1. The box will only accept valid passwords

1.1.9.1.1. A password must be at least 8 characters long

1.1.9.1.2. A password must contain at least one capital letter

1.1.9.1.3. A password must contain at least one lowercase letter

1.1.9.1.4. A password must contain at least one special character

1.1.9.2. The user input will be hidden using *

1.1.9.3. The box will be marked as required using *

1.1.10. A label for Verify Password

1.1.11. An input box for Verify Password

1.1.11.1. The user input must be the same as the input in the Password box

1.1.11.1.1. If the input is not the same, the user will be prompted with a message in a display box

1.1.11.1.1.1. The message will say “Passwords do not match”

1.1.11.1.2. If the input is the same, no message will be displayed

1.1.11.2. The box will only accept valid passwords

1.1.11.2.1. A password must be at least 8 characters long

1.1.11.2.2. A password must contain at least one capital letter

1.1.11.2.3. A password must contain at least one lowercase letter

1.1.11.2.4. A password must contain at least one number

1.1.11.2.5. A password must contain at least one special character

1.1.11.3. The user input will be hidden using *

1.1.12. A button labeled “Create Account”

1.1.12.1. On click, the email is checked to make sure an account with that email has not already been created

1.1.12.1.1. If there is an existing account with that email, a message will be shown in a display box

1.1.12.1.1.1. The message will say “Email already taken”

1.1.12.1.2. If there is no existing account, no message will be displayed

1.1.12.2. If any of the fields are left empty, a dialog box will show containing the message, “Please fill out the required fields”

1.1.12.3. If all required fields are filled out, no message will be output

1.1.12.3.1. All information will be submitted to the back-end servers

1.1.12.3.2. The user will be logged into the account

1.1.12.4. The application will get sent to the home page

2. Sign-in Form

2.1. The sign-in form will be one component with the following:

2.1.1. A button labeled “Sign In”

2.1.1.1. On click, this button will send the user to a sign in form

2.1.2. A label for an email address

2.1.3. An input box for the email address

2.1.3.1. The user input will be verified by the server

2.1.3.2. If the user submits an unverified email, a dialog box will pop up that states “Could not find the email”

2.1.3.3. If the user submits a verified email, no dialog box will pop up

2.1.4. A label for a password

2.1.5. An input box for the password

2.1.5.1. The box will only accept valid passwords

2.1.5.1.1. A password must be at least 8 characters long

2.1.5.1.2. A password must contain at least one capital letter

2.1.5.1.3. A password must contain at least one lowercase letter

2.1.5.1.4. A password must contain at least one number

2.1.5.1.5. A password must contain at least one special character

2.1.5.2. The user input will be hidden using *

2.1.5.3. The user input will be verified by the server

2.1.5.4. If the user submits a unverified password, a dialog box will pop up that states “Incorrect password”

2.1.5.5. If the user submits a verified password, no dialog box will pop up

2.1.5.5.1. The user will be directed to the “Home” page

3. Dialog Box

3.1. Dialog box will have an “Okay” button that will close the dialog box

3.2. Dialog box will provide and explain the error

4. Navigational Bar

4.1. The navigation bar will consist of the main components for teachers and administrators:

- 4.1.1. The Home button
  - 4.1.1.1. On click, this button will take the user to the “Home” page
  - 4.1.1.2. This section will display the upcoming scheduled events
- 4.1.2. The Roster button
  - 4.1.2.1. On click, the Roster button will take the user to the “Roster” page
  - 4.1.2.2. Teacher
    - 4.1.2.2.1. On click, the user will be sent to their student list
  - 4.1.2.3. Admin
    - 4.1.2.3.1. On click, the user will be sent to a list of teachers
- 4.1.3. The Visiting Students button
  - 4.1.3.1. On click, the Visiting Students button will take the user to the “Visiting Students” page
- 4.1.4. The Account button
  - 4.1.4.1. On click, the Account button will take the user to the “Account” page
- 4.1.5. Contact Information button
  - 4.1.5.1. On click, the Contact Information button will take the user to the “Contact Information” page
- 4.1.6. Schedule Event button - Admin Only
  - 4.1.6.1. On click, the Schedule Event button will take the user to the “Schedule Event” page
    - 4.1.6.1.1. On this page, the user will be taken to a form to schedule a date and time for an event

5. Roster

5.1. The Roster page will consist of three main components:

- 5.1.1. A dropdown field that has the class name for selection
  - 5.1.1.1. Each class is its own selection of students and allows multiple periods that a teacher might be teaching.
  - 5.1.1.2. The last class selected is remembered for future page loading.
- 5.1.2. A “Submit” button
  - 5.1.2.1. On click, the roster will get submitted to the admin

6. Visiting Students

6.1. The Visiting Students page will consist of one main component:

- 6.1.1. A comment box
- 6.1.2. A submit button
  - 6.1.2.1. On click, the comment box contents will be sent to administrators

7. Contact Information

7.1. The Contact Information page will consist of two components:

7.1.1. The phone number for the administrator

7.1.2. The email for the administrator

8. Scheduled Events - Teacher Only

8.1. The Scheduled Events page will consist of one component:

8.1.1. Date and time of the scheduled event

8.1.2. If no events with Date being current or future shows, the screen will show “No Scheduled Events”

8.1.3. Past events will be shown in table below that allows teachers to pull up rosters of past events

9. Reports - Administration Only

9.1. Reports will consist of one main component:

9.1.1. Links to each teachers roster and attendance

10. Archives - Administration Only

10.1. Archives will consist of one main component:

10.1.1. A list of dates where reports have been submitted

10.1.1.1. On date click, the administrator will receive a report from that day

11. Feedback

11.1. Feedback will consist of one main component:

11.1.1. A comment box

11.1.2. A submit button

11.1.2.1. On click, information from the comment box will be sent to the administration

12. Dashboard

12.1. Show events list that in a scrollable view

12.1.1. Admins can make events and change events in the admin panel

13. Admin Panel

13.1. This Admin Panel will be made up of multiple components consisting of:

13.1.1. Shows table with headers of, (Last Name, First Name, Email, classes assigned)

13.1.1.1. Each row has Last Name, First Name, and Email all of which are editable

13.1.1.1.1. If an row is edited an module will pop up asking if the user wants to commit the changes

13.1.1.1.1.1. Module will have a submit button that saves changes, and a cancel button which will close the module.

13.1.1.1.1.2. When canceled the changes will revert on screen.

13.1.1.2. Classes assigned is in a module, that opens up a new table of classes

13.1.1.2.1. Each class will have a separate table with header showing class name, and table headers

13.1.2. A Password Reset button

13.1.2.1. Shows table with headers of, (Last Name, First Name, Email, Button)

13.1.2.1.1. On click, the page will show the module

13.1.2.1.1.1. The page will show the email that the password reset will be sent to, and will inform the user of a randomly generated password to sign in

13.1.2.1.1.2. This page will show a submit button to send and reset password

13.1.2.1.1.3. A cancel button

13.1.2.1.1.3.1. On click, the module will be closed out

13.1.3. Events Page

13.1.3.1. On click, the Events page will show a “create event” form on top with table of previous events below

13.1.3.1.1. Date and time will at first be set to current time but can be changed

13.1.3.1.2. Label saying Event Name

13.1.3.1.2.1. Name of event input box

13.1.3.1.3. Label Description

13.1.3.1.3.1. Text Input Box.

13.1.3.1.4. Submit button

13.1.3.1.4.1. If within 15 mins or now

13.1.3.1.4.1.1. Shows modal informing that a push notification will be sent to all users that an event is being at specified time, do you want to send push notifications?

13.1.3.1.4.1.2. Send button that sends.

13.1.3.1.4.1.3. Don’t send button to not.

13.1.3.1.4.2. If an event has been made within the past 15 mins shows a model saying an event has been posted that occurred in the past 15 mins.

13.1.3.1.4.2.1. Checks in place that scheduled events are not accounted for.

13.1.3.2. Below create event shows a calendar implementation and a table that allows administration to input description

13.1.3.2.1. The short desc is there to allow the first 180 characters to show from the desc of the event.

13.1.3.2.2. The Open button will show a model that shows all information on Time, Event Name, Description, and what account made the event.

13.1.4. Super Admins User page

13.1.4.1. Shows table with headers of, (Last Name, First Name, Email, is admin?, buttons)

13.1.4.1.1. Each row has Last Name, First Name, Email, remove button

13.1.4.1.1.1. On click, a module with confirmation will show asking to confirm.

13.1.4.1.2. Last row will be empty that input can be entered

13.1.4.1.2.1. Email will have same validation as registration

13.1.4.1.2.2. Submit button to create an admin.

13.1.4.1.2.2.1. When submitted an email with a randomly generated password will be sent.

14. Chat Panel

14.1. Shows as a widget that provides a chat for all teachers and admins.

14.1.1. General chat will be in slow mode in which there is a 20 second cooldown between chat messages, admins will bypass this cool down.

14.1.2. People can Mention people. Mentioned people will show as highlighted and send push notifications.

Updated Timeline

See **Figure 3** in the appendix for a detailed description.

Appendix

Figure 1: Team Member Names

Name	Role
Katie Hoskins	Developer
Tanya Peacock	Developer
Marcus Strange	Developer

Figure 2: Tools and Technologies

Place in System	Dependency Name	Dependency Type	Why?
Frontend	Vue.js	Frontend	Vue.js is a collection of premade web dev templates and uses Nuxt.js which is built on top of React
Data Source	Supabase	Database	We need a strong database for our project and this seems like the most reasonable one to use for our needs.

Figure 3: Updated Schedule

Week 1: August 28 - September 1	**All Members:** Project Proposal, discovering tools and technologies to use for the project, discussion about requirements, meeting with team and discussing strengths of team members.
Week 2: September 4 - September 8	**Sep 4: Labor Day - No class** **All Members:** Finish project proposal, discover requirements for the project, continue talking about tools and technologies available for the project and think about requirements.
Week 3: September 11 - September 15	**All Members:** Work on and finish requirements document, collaborate on what would be possible for the project, work on and finish requirements powerpoint.
Week 4: September 18 - September 22	**Katie:** Look into and learn more about the tools and technologies being used including Vue.js, Nuxt, React, and Supabase, start tinkering around with the technologies. **Tanya:** Research and learn more about the resources

	we are looking at using. **Marcus:** Provide insight in the tools and teach core practices in building the systems needed in the project.
Week 5: September 25 - September 29	**Katie:** Work on making website mobile friendly and working on the correct application, get GitHub created. **Tanya:** Start on the Registration Form. **Marcus:** Start on Registration backend, and setup the proper services in Supabase
Week 6: October 2 - October 6	**Katie:** Navigation bar, some frontend design. **Tanya:** Polish the Registration form, make sure all the buttons work like they should and labels line up with boxes. **Marcus:** Write out the documentation on Registration and start on admin controls with some frontend design.
Week 7: October 9 - October 13	**Oct. 9-10 - Fall Break No Class** **Katie:** Working with roster set up for teachers and designing classes for the teachers. **Tanya:** Learn about how to use the database and get the registration form to save to where it needs to. **Marcus:** Setup teacher’s table with class structure.

Week 8: October 16 - October 20	**Katie:** Continue work on the roster setup for teachers. **Tanya:** Assist where needed. **Marcus:** Setup class's table with student's structure
Week 9: October 23 - October 27	**Katie:** Work in some frontend design for the website including making clickable icons and making it user friendly. **Tanya:** Assist where needed. **Marcus:** Setup student's table and setup rules for creating/editing students.
Week 10: October 30 - November 3	**Katie:** Help work on admin panel and discuss what it should be used for. **Tanya:** Assist where needed. **Marcus:** Setup process rules for events. Possibly complete admin process.
Week 11: November 6 - November 10	**Katie:** Continue work on admin panel and communication with teachers. **Tanya:** Assist where needed. **Marcus:** Help implement push notifications of events.
Week 12: November 13 - November 17	**Katie:** Work on archives for previous reports being sent to administration. **Tanya:**

	Assist where needed. **Marcus:**
Week 13: November 20 - November 24	**Katie:** Continue work on archives, start finishing up code and polishing frontend items. **Tanya:** Assist where needed. **Marcus:**
Week 14: November 27 - December 1	**All Members:** Finish up things we are unclear about and work on the project board.
Week 15: December 4 - December 8	**All Members:** Clean up code and finish submitting all items, finish project board.
Week 16: December 11 - December 15	**Finals Week** **All Members:** Present our fully functioning project.

This timeline is subject to change. *