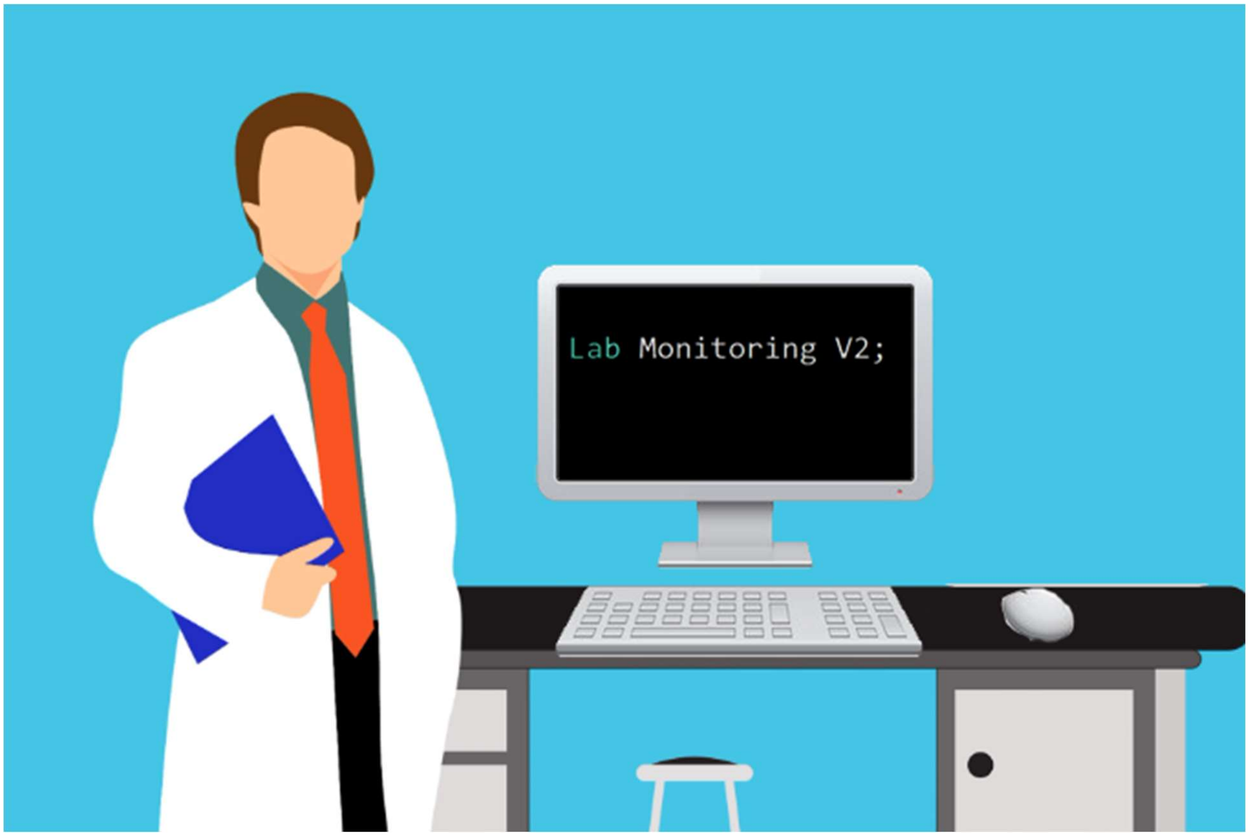

# Lab Monitoring V2

04.01.2024

—

## Team: RSR

Ryan Burns

Spencer Roucoulet

Ryan Plante

## Problem

- The lab monitoring website can be inconvenient for practical use. Converting it to a standalone mobile app will allow for easier day to day use.
  - Website isn't formatted for mobile and is only user friendly on a desktop screen.
  - Since it's an aspx project, we haven't figured out a way to deploy the current site so you need to build the project to run the website.
- Some students with COVID or other illnesses are unable to get their tutoring needs met since tutoring is in-person only.
- Reporting on borrowed items is inconsistent and hard to use.
- Student IDs need to be manually entered. Would be nice to scan the barcodes on the IDs.
- Export to pdf doesn't work for reporting
- No ban system if there's a troublesome student.
- Monitors/tutors have to self-report their schedule to their respective department head manually, causing a delay in working hours for the monitors and it's harder for the department heads to generate schedules without collisions in schedules.
- Admins can accidentally promote students by accident to department heads because the user management system isn't robust

## Users

1. Students
2. Lab monitors/Tutors
3. Department Head/Admin

## High Level User Requirements

1. **Students** will need to check the lab schedule and perhaps chat with a tutor so they can get quick help with tutoring or set up a private tutoring session. Not all students have a computer so it would be beneficial to have a mobile app for the user.
2. **Lab monitors** will need a more robust way of checking in students and checking out items from the lab. Not all lab monitors will have a computer so it would be

convenient to have a mobile app and add scanner functionality to streamline the check in process by scanning student IDs. The scanner functionality can also be used to check in and out items that students borrow from the lab so that these items are tracked. There needs to be a way to differentiate lab monitors and tutors as some lab monitors might not be a tutor and vice versa. But sometimes lab monitors work as a tutor, which was the case with Ryan P and Spencer R.

3. **Tutors** need the ability to chat with students through a chatroom so that they can offer private one on one sessions or appear in a chatroom for any student to join during their working hours. It will provide notification sounds for users joining and sending messages.
4. **Department heads** will need the ability to use desktop / mobile apps. Some department heads prefer the old school way of doing paperwork, reporting, etc. from the computer, but some department heads might prefer the convenience of a mobile app and want to download it on their phone. Department heads will need to generate reports done on items borrowed and students that entered the lab. Since the lab monitor/tutor positions are work study positions, department heads need a better way to create and modify schedules for the student while keeping in mind the more restricted availability caused by the monitors being students attending classes at NEIT.
5. **Admins** will need the ability to manage user accounts. This includes full CRUD functionality of the users table. Unless this project gets officially approved by the school to integrate with their existing systems, this is necessary to modify user accounts because NEIT probably uses Microsoft's sign-on APIs to manage apps on the school's intranet. They also need the ability to modify departments and promote/demote department heads.

## Technology Used

- C#
- .NET MAUI
- SQL Server Management Studio
- socket.IO

# High Level Database Architecture

## Users table

- A table storing all of the users who can log into our program
- Store their name, encrypted password, salt value, department ID, privilege level, and position information (whether they are a monitor, tutor, or both), alongside a simple boolean keeping track of whether or not they are a teacher as a safeguard to prevent the promotion of the wrong users

## Department table

- Store the department ID and name
- Used for linking users and labs to departments

## Lab table

- Store lab name, room, and the department ID tied to them
- When displaying lab information, the department head can be used to pull the department tied to the lab

## Log table

- Store studentID, time in, time out, lab ID
- For the comings and goings of students working in the open lab

## Schedule table

- Store userID, department ID, and two csv-formatted text schedules that will be parsed by the program - one entered by the monitor/tutor to show their availability, and another for their shift schedule

## Items table

- Store description, quantity, and serial number
- Item id will be used for bar/QR codes

## Item log table

- Store transaction type (in or out), item ID, a timestamp, the ID of the student using the item, and the ID of the monitor checking it out

## Permission lookup table

- Table to see which primary key value corresponds to which level of access - 0 for student, 1 for monitor/tutor, 2 for department head, 3 for admin

## Error log table

- Stores information about errors - log type (informative, error, warning, critical), timestamp, description, stack, source, exception type, user ID

## Audit log table

- Store a timestamp and a description of the action taken - i.e. “User 008015394 inserted record 45 into log”