Progress Notes

Timeline and Workloads:

Work Breakdown

Roles

1. **Developer 1: Database and ETL (Pablo)**
- Set up SQL Server and create the database schema.
- Implement the ETL process to import data from `CY22.xlsx`.
- Ensure database connection configurations are functional.
2. **Developer 2: Backend Logic (Anthony)**
- Implement model objects (`Company`, `Incident`, `Railroad`, etc.).
- Develop methods for data retrieval and user management.
- Set up logging of database commands in JSON format.
3. **Developer 3: User Interface (Connor)**
- Create the UI for login, main screen, and data management.
- Implement data grids for incidents, companies, and railroads.
- Handle user input validation and error messages.

Timeline

Week 1: Planning and Initial Setup (Days 1-5)

Day 1: Kick-off meeting to discuss project details and expectations.

Day 2:

- Developer 1: Set up the SQL Server environment.
- Developer 2: Draft the model objects and database schema.
- Developer 3: Design initial UI layouts (wireframes).

Day 3:

- Developer 1: Finalize the database schema.
- Developer 2: Implement model classes in C#.
- Developer 3: Create mockups for the login and main screen.

Day 4:

- Developer 1: Begin work on the ETL process.
- Developer 2: Implement database connection configurations.
- Developer 3: Start UI development for the login form.

Day 5: Team sync-up to discuss progress and blockers.

Week 2: Development (Days 6-10)

Day 6:

- Developer 1: Finalize ETL functionality.
- Developer 2: Implement logging functionality.
- Developer 3: Develop main screen UI.

Day 7:

- Developer 1: Test and validate the ETL process.
- Developer 2: Work on data retrieval methods.
- Developer 3: Continue development of the incident form.

Day 8:

- Developer 1: Ensure all database commands are logged correctly.
- Developer 2: Implement user authentication logic.
- Developer 3: Complete the incident form and start search functionality.

Day 9:

- Developer 1: Conduct testing of the database setup and ETL.
- Developer 2: Finalize model methods and user maintenance features.
- Developer 3: Finalize search functionality and implement error handling UI.

Day 10: Team sync-up to review progress and integration plans.
Week 3: Integration and Testing (Days 11-13)

Day 11:

- All developers integrate components and resolve any conflicts.

Day 12:

- Conduct thorough testing of the application, focusing on functionality and logging.

Day 13:

- Finalize documentation: configuring database connections, default config, and project artifacts.
- Prepare for submission, ensuring all components work smoothly.

Progress Notes

- Maintain a shared document for progress notes, updating it at least every two days.
- Use project management tools (like Trello or GitHub Issues) to track tasks and responsibilities.

Progress Notes

Day 1:

- Meeting to go over who is doing what.
- We talked about the timeline and gave everyone their roles for the project.

Day 2:

I made 7 form designs, will continue making them through Day 3.

Day 3:

Finished up the form designs. Will make changes, if need be, later down the line but we have the outline done.

Day 4:

Researching importing data and trying a few ways out. Set on one to start the next day.

Day 5:

Finished up one script to import data.

Day 6:

Finished up the rest of the scripts to import data. Created user table in the db, working on db connection class, user class salting and hashing, and trying not to kms. Added UserSeed to populate db with default user, as well as populating the db with it as well. Created startup class for too the initial user seed, called this in the program.cs. added a databasse connections class that handles the db connection, added users class that handles and stores the password info. put login logic for the login button, which checks the username, tries to authenticate password to make its correct

Day 7:

Worked on a non functional script and hating life.

Day 8:

Gave up on the script and decided to start making the forms functional and got the main homepage form working properly (buttons, data table, username showing up). Plan to work on a few more forms tomorrow.

Day 9:

Anthony- have been working on getting the connection to gather data for those who logs in and out. Ran into multiple errors and BD was down for a bit. Have code ready but need to work out errors.

Connor: Got a few more forms functional. Got the data to load into the form on startup and making buttons do what they are supposed to do.

Day 10:

Anthony- Deleted all new code that i replace with the old. Going back to my first check point and will be working on logic (making queries for Visual studios so it can pull in information on who logs in and out.) This should get it working. (For the love of GOD)

Connor: Redid the data in the database to make sure only one ID is present for each company, railroad, and train ID. Figured out the double click function in the forms.

Pablo: Fixed userId logic for the log events, finished insert new incident form (going to kms)

Day 11:

Fixed the data scripts one last time and worked on fixing the logic behind the one double click form. We should be finished with the project now and will be working on final documentation as well.

Day 12:

Worked on final documentation

Final Documentation

GitHub Repo: https://github.com/ConnorH11/CNSA-212-FINAL.git

Importing Data:

To import the data, we made four PowerShell scripts and a C# console app to do it. In the "Add Data" folder in the repo, there are four PowerShell scripts. They are named by what table they are populating with a number 1-4 at the end. That number indicates what order you will need to run those scripts. You can run these scripts just like any other Powershell scripts. The last one for the incidents table is under the Script folder. You can do DOTNET Build and DOTNET run. After that all the data should be imported into the database.

Running the Forms:

As stated in the readme form, you will have to run the program under the CNSA-212-Final folder. It will open the database connection form where you just need to put the IP address "100.84.117.15" in and click connect. You will then need to rerun the program to bring up the sign-in page. You can use the account with the username "admin" and password "CNSAcnsa1". After logging in, it will take you to the main form which will have some basic data pulled up on the data grid view and buttons to take you to different forms. Click the "Show Incidents" button to take you to a new form that will populate data from the incidents table. If you double-click on the SEQNOS number, it will open a form that shows you more information on that incident. If you click "Show Companies" it will do the same thing as the incidents but show the companies' table. If you double-click on a company name, you will see a new form that shows all of the incidents a company has been involved in. Railroads are pretty much the same way as companies. If you click "Insert Incident" it will take you to a form that has entries for each column in the database and when you insert the data, it inserts into the DB. Database Configuration is the same form that shows up when you run the program the first time. Adding a new user is kind of like an insert incident. Changing the password is for you want to change the password. Log Event is the login and logoff logs.