

# CPSC 408 Final Report

- I. [Problem](#)
  - II. [Related Work](#)
  - III. [Elements of Solution \(Framework\)](#)
  - IV. [Results & Schema Diagram](#)
- 

## Problem

With my work as a teacher, I realized that my workplace's current method of keeping track of timesheets and pay is a bit inefficient. They utilize Google sheets and have everyone on the same sheet, making it so everyone's time is open to others. The formatting of the Google sheet is a bit inefficient as well since employees have to either calculate their own hours by hand or search up and copy and paste a formula every time. The issues presented with utilizing a document based database for timesheets definitely highlights the need for a potential app that could securely allow employees to enter their hours without seeing others, and allow do calculations for the employee instead of by hand.

## Related Work

Related work to my project could include any other timesheet system that companies or businesses may use. In previous work, I have used apps that allowed a check-in/check-out process by using your location and automatically logs your hours, calculates your pay, and even message your employer when needed. My app will be a prototype of their design with basic timesheet needs but the theme is centered about timesheets for teaching so it will include students, schools, teacher information, etc.

## Elements of Solution (Framework)

My solution's framework included different menus (or screens) that allow the user to make options based on their ID number. A user would log in with their ID and then would be able to check out their time, students, money, and information. With the time menu, users are able to enter their hours as entries. They can also delete hours if they make a mistake and would like to fix their sheet. Users can also check their timesheets and previous hours in this menu. Moving onto the students menu, there is only one option that just prints a list of every student and their grade to the user. This allows the user to check who is in their class. In the pay menu, the user is able to either calculate their pay or export their pay. When calculating their pay, the program will ask for their pay rate per hour and then find the product of the difference of their hours to calculate their next pay. The user can also export here. The program will ask for a file name and either write to that file or create a new one and export the user's hours and their predicted pay to a CSV file. In the information menu, the user is able to print out their information (Name, School) or update their school name.

## Results & Schema Diagram

The results of my program implemented the required functionality, but did have some minor mishaps. Initially, the program would have had a better interface in TKINTER or Swift but due to time constraints, the program was ultimately finished in command-line. However, it does still implement all of the functions as stated in the framework and works the way it was intended. In the future, possibly adding more features such as adding a user to system, creating a manager

account that can edit everyone's timesheets, and more descriptive tables would enhance the program. Attached below are the schema and ER diagrams for my project.

