**Software App Wizards**

**SDEV-120 Group Project**

**Payroll Management system**

**Submitted by Samuel Steinhardt (et al.)**

Updated Project Plan – Software App Wizards (SAW)

## 1. Team Leadership

The current Team Leader is Samuel Steinhardt. Each Monday after SDEV class, the team will decide whether to rotate leadership. If no decision is made, Samuel Steinhardt defaults to Team Leader for that week.

## 2. Communication & Collaboration Tools

Sofware App Wizards has been using github and discord to log attendance and check in.

## 3. Team Identity

Our team, The Software App Wizards, are all about maintaining the human element in coding, It is the magical secret sauce that see’s projects through to the end and enables great achievement.

## 4. Programming Approach

Started with flowcharting but later gravitated to psuedocode, as everyone was more of a fan and were better able to communicate ideas.

**5. Security & Validation Approach**

* **Input validation:**
  + Hours must be **≥ 0**; values
  + Dependents must be **≥ 0** (integer).
  + Employee **ID must exist** in the rate database before any calculations.
* **Calculation rules (pre-tax basis):**
  + **Gross pay** = up to 40 hours at base rate **+** hours over 40 at **1.5×** the rate.
  + Taxes are computed on **pre-tax (gross)**: **Federal 7.9%**, **State 5.6%**.
  + **Net pay** = gross − (federal + state).

## 6. Attendance Tracking

Attendance has been tracked in weekly discord calls. Group meets every monday after the SDEV 120 lecture to discuss goals and progress for the week.

## 7. Initial Payroll Program Plan

The project’s objective is to build a payroll program that calculates employee wages, applies tax rules, and outputs results. I have asked the professor about getting extra credit. He has agreed on some amount of extra credit for coding the modules in python. If this can be achieved is TBD and not mandatory for anyone. The assumption is that we will be using psuedocode to code modules and that the structural logical result should be a program that calculates a payroll and applies taxes.

- Inputs: Employee work hours, hourly rate, constant data such as income tax

- Processes:

## 1. Initialize constants and income tax (Housecleaning)-Alexia Erkman

## 2. Get employee input (hours and rate)-Samuel Steinhardt

## 3. Run calculations (gross pay, deductions, net pay)-Jayden Johnson

## 4. Output payroll details for each employee-Wai Moo

## 5. End of program with a final “End of Job” message-Alexia Erkman

- Outputs: Gross pay, deductions, net pay, and an end-of-job report

- Modules: Housecleaning, Main Code Loop(Input, Calculation/Run, Output), and Termination(End of Job)

## 7. Team Member Participation

- Samuel Steinhardt – Participated and led the team to the completion of the project

- Wai Moo – Aided in pseudocoding modules through discord calls

- Jayden Johnson – Wrote exporting to excel spreadsheet module and made excel spreadsheet test cases

- Alexia Erkman – Coded housekeeping and end of job logic.