BigProject: Time and Attendance System

Your goal is to create a system to help supervisors manage time and attendance for employees. What follows is a brief introduction to the business case, a recommended set of UML diagrams for a system, and a description of the required attributes for your system.

The business case

Joe-Bob's Burgers (JBB) creates heavenly burgers using unusual ingredients. You never know what Joe-Bob will dream up next. From his Kangaroo and Mint to his Mushroom, Bleu Cheese, and Pineapple concoction, JBB fills a void that nobody knew was even there until they have tasted the quality.

Joe-Bob will tell anyone who will listen that his secret to success is to make every burger with love. But since he doesn't make every burger himself, he strives to take great care of his employees. From a $20 per hour minimum wage to a generous paid time off policy, Joe-Bob really tries to do right by his employees.

When JBB was a young business, this was not that big of a deal. They used manual timecards and the supervisors kept an Excel Spreadsheet of all sick and vacation time earned, accrued, and used. But JBB has been wildly successful ever since the introduction of the Varsity Burger, a longhorn burger with a maroon bun. It seems every good Aggie likes to eat Varsity after its horns were sawed off? This led to great opportunity but new complexity. JBB now has over 120 locations in Texas alone, and is looking to expand across the nation. For this to work, Joe-Bob needs a reliable and fast time and attendance system.

Each JBB location has a general manager that manages all of the other employees at that location. No employee whose time and attendance will be tracked in this system can work for more than one location. There are also shift managers and assistant managers that will help enter information into this system.

Employees earn 2 hours of sick time and 2 hours of vacation time each week that they work at least 5 hours. Assistant Managers earn 2 hours of sick time and 3 hours of vacation time each week that they work at least 5 hours. Shift Managers earn 2 hours of sick time and 4 hours of vacation time each week that they work at least 8 hours. Store managers earn 4 hours of sick time and 8 hours of vacation time each week *that they are paid for* at least 38 hours.

Vacation time can be used within 2 years of earning it. After 2 years, the time expires and the employee is paid for the time. Sick time can be used within 1 year of earning it. After 1 year, the time expires and the employee is not paid for it; however, Joe-Bob likes to keep track so if someone has an emergency and has given up a lot of time in the past, he will just pay the employee anyway even though they don't technically have the hours available. Vacation and sick time can only be used in one-hour increments.

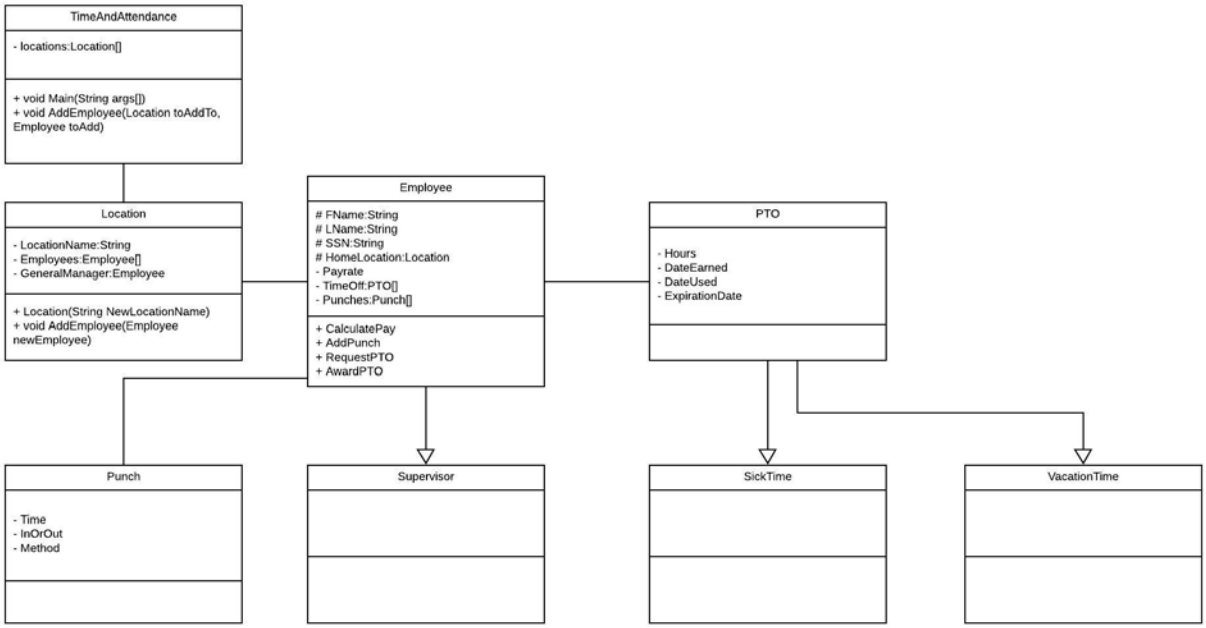
All employees clock in and out, even managers. If (when) they forget to clock in or out, some type of manager must create the clock in or clock out record for them. No manager can create a clock in or clock out for himself or herself.

All timecards must be approved by a shift manager, except the general manager must approve all shift manager and his or her own timecard. An unapproved timecard will not be paid but will remain in a pending state.

Paychecks are generated weekly. The taxes and withholding is figured in another system. All that the time and attendance system is expected to do is output the hours worked, the hours of sick leave used, the hours of vacation time used, and the number of hours of sick time and vacation available for each employee.

Recommended Solution

What follows are UML diagrams to help you get started solving this case. You are NOT required to use all of the classes and CAN change any method names you desire. You will need to add methods and variables to make your solution work, and that's OK. I am only giving you a starting point!



Required Attributes

Your solution must:

1. Read in a file passed in as a command line argument AND read in a file (or multiple files) entered from a menu.
2. Have a menu action to allow a user to search for a record that matches one or more parameters provided by the end-user at runtime.
3. Have a menu action to sort all records based on one or more parameters provided by the user at runtime.
4. Use an appropriate data structure to hold necessary records for processing.
5. Use polymorphism for at least one aspect of the program.
6. Have a menu action to add a record.
7. Have a menu action to change a record.

8. Have a menu action to delete a record.
9. Have a menu action to save all data to a file (or multiple files).