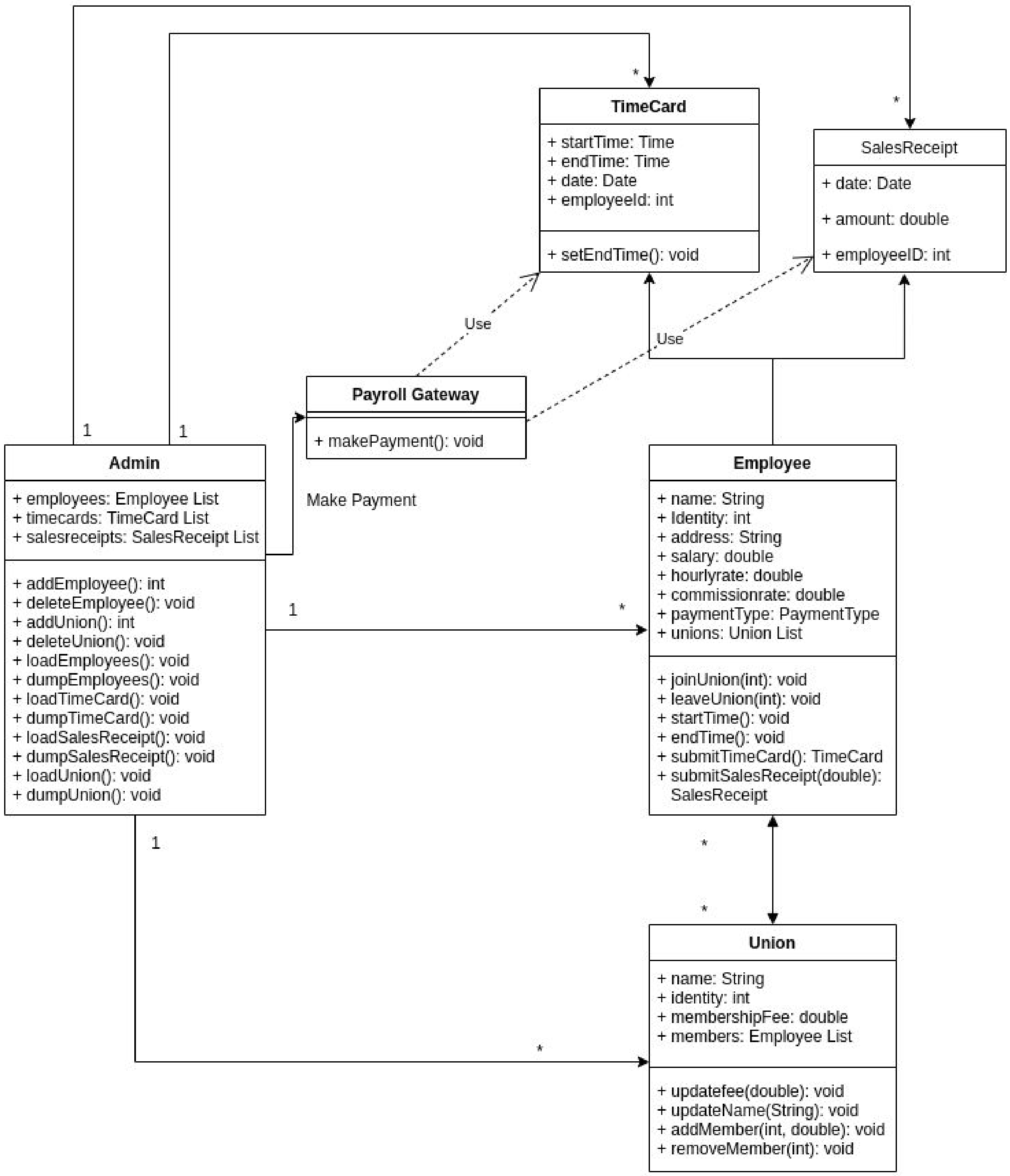
Class Diagram



Documentation

Requirements:

Objective: To design a Payroll management system with following features:

1. Addition and Deletion of employees should be possible.
2. Option of mulitple modes of payment for the employee.
3. Employees can be salaried or be hired on a per hour basis or maybe both.
4. Employees get overtime for working more than 8 hours a day.
5. Employees get sales commission for the amount of sales they make.
6. Options to add and delete Union. Employees can join some union. Employees have to pay there membership charges and Some extra miscellanious charges that union ask for.
7. These charges has to be deducted from the next salary of the union member.

Design:

I have tried to create my design toward SOLID principles. I have based my design towards Single Responsibility and Loose coupling. Though there are much improvement possible which I could not incorporate because of lack of time. I have also tried to keep the naming convention of variables as easy to understand as possible. Here is the abstract of the work performed by different entities:

1. **Admin(Manager):**
    a. Manager is responsible to add and delete employee, create and delete union.
    b. Manager is also working as a database container in the case. It has access to the database(in the form of json) and is responsible to load data from database and dump back into it.
    c. Manager has the key to do make payment(Yet to be implemented).
2. **Employee:**
    a. Employee can change it's data.
    b. Employee can decide to join or leave an union.

    c. Employee can create a TimeCard at the start of the day and submit it at the end of the day to the manager.
    d. Employee can create SalesReceipts based on the sales it has done and then submit it to the manager.
3. **Union:**
    a. Change it's charges.
    b. Create new miscellaneous charges.
4. **Payment Gateway:**
    a. It moves through all the TimeCards and SalesReceipts and pay the employees according to it.

The above objects are mostly independent and changing one doesn't require others to change. This makes sure of the Design towards Single Responsibilty and Loose Coupling. I have also created the Admin, Employee and Union object based on interfaces. It will help in extending the project further if the requirement changes and some new interface is needed to be added.

Future Improvements

1. Integraion of database for easy handling of data.
2. Creating multiple interfaces for the employee based on the type of salary they are taking.
3. Give more options to union to add additional charges and policy for the employees.
4. Give option to the employee to change the information.
5. Handle payment option. Still to implement.

P.S. Couldn't finish completely because, electricity got cut after a storm at evening. Laptop and inverter got discharged. Finished the report after long sleepless night after in the morning. Will try to finish it today.