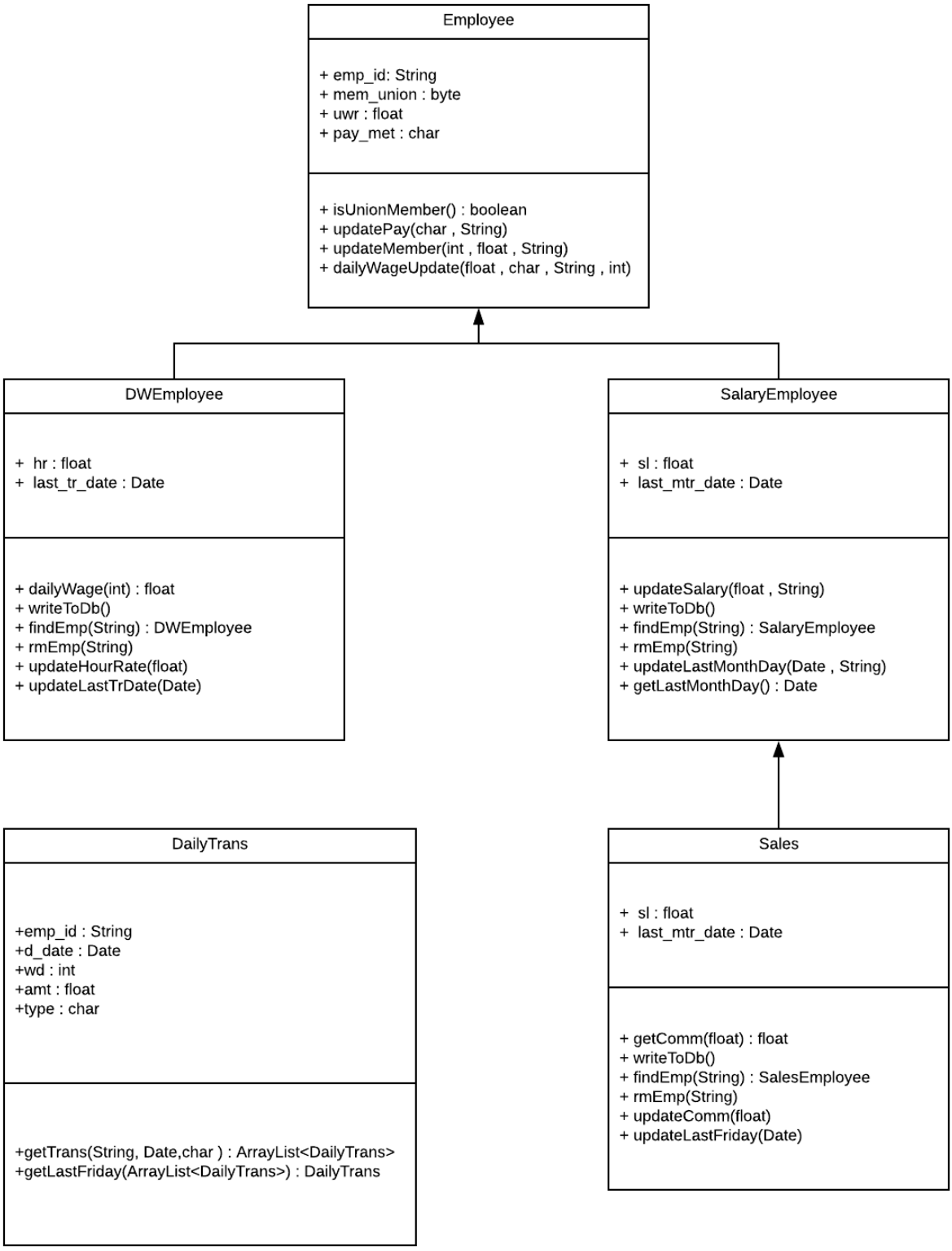
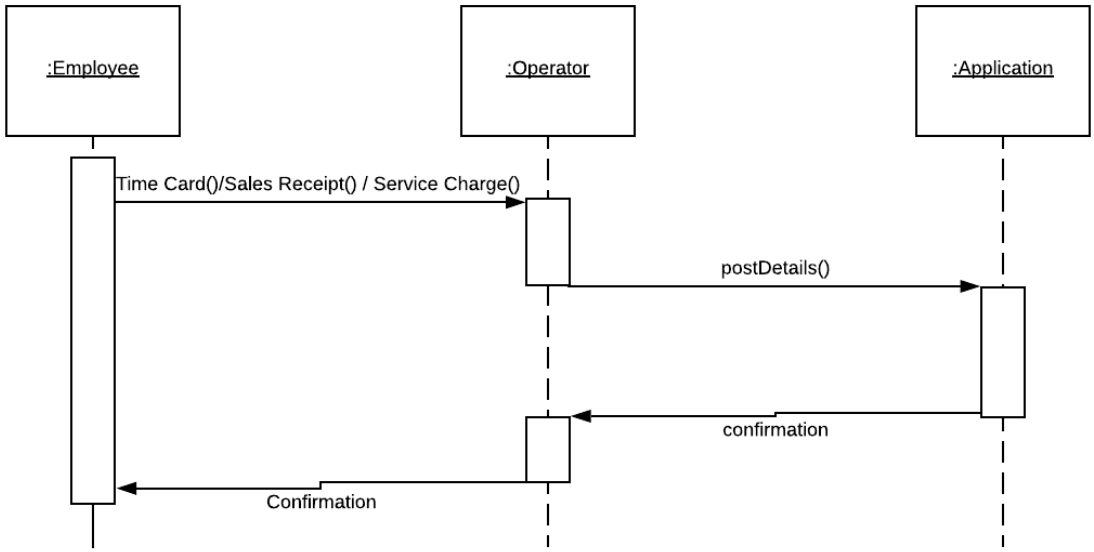
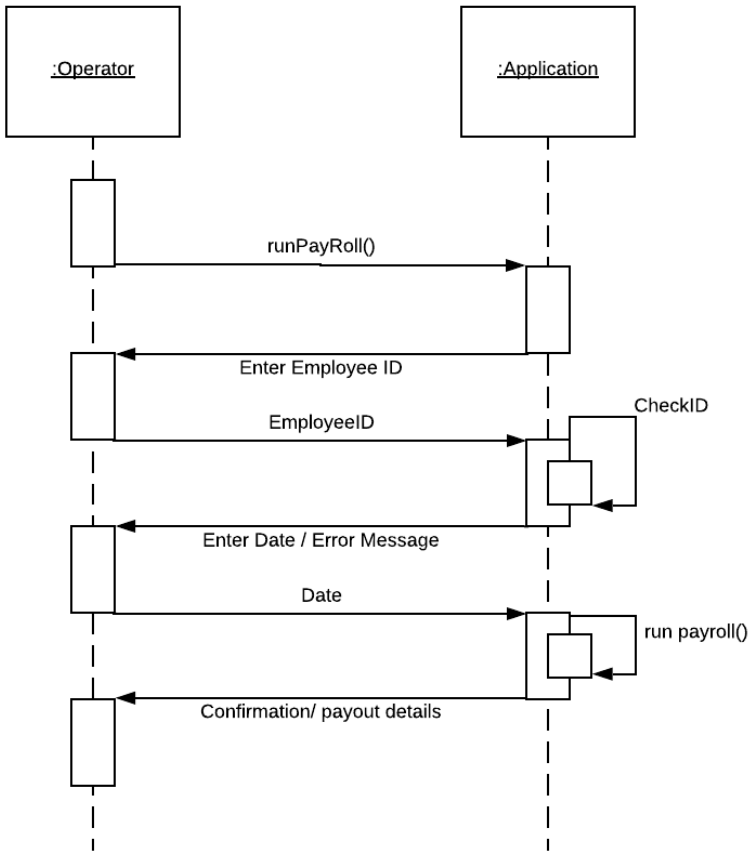

Employee
+ emp_id: String
+ mem_union : byte
+ uwr : float
+ pay_met : char
+ isUnionMember() : boolean
+ updatePay(char , String)
+ updateMember(int , float , String)
+ dailyWageUpdate(float , char , String , int)
DWEmployee
+ hr : float
+ last_tr_date : Date
+ dailyWage(int) : float
+ writeToDb()
+ findEmp(String) : DWEmployee
+ rmEmp(String)
+ updateHourRate(float)
+ updateLastTrDate(Date)
SalaryEmployee
+ sl : float
+ last_mtr_date : Date
+ updateSalary(float , String)
+ writeToDb()
+ findEmp(String) : SalaryEmployee
+ rmEmp(String)
+ updateLastMonthDay(Date , String)
+ getLastMonthDay() : Date
DailyTrans
+emp_id : String
+d_date : Date
+wd : int
+amt : float
+type : char
+getTrans(String, Date,char ) : ArrayList<DailyTrans>
+getLastFriday(ArrayList<DailyTrans>) : DailyTrans
Sales
+ sl : float
+ last_mtr_date : Date
+ getComm(float) : float
+ writeToDb()
+ findEmp(String) : SalesEmployee
+ rmEmp(String)
+ updateComm(float)
+ updateLastFriday(Date)

## FK PAYROLL POST




## FK PAYROLL FUNCTION



# FlipKart PayRoll Design

## EMPLOYEES:

Following Details are stored for employees in Employee class :

- Employee Id : 6 Character long String
- Union Member : Byte , Belongs to Union or Not
- Payment Method : Character , 'A' : Postal Address , 'P' : Paymaster , 'B' : Bank Account

Employee Class is further subclassed into DailyWageEmployees , SalaryEmployees, SalesEmployees which store details like hour_rate , sales_commision , salary etc.

SalesEmployee is a sub class of SalaryEmployee .

The basic details of Employees are stored across three tables in database : **dailywageemployee , salesemployee , salaryemployee.** The emp_id is the primary key in all tables.

**DESIGN PROBLEM 1 :** the application can allow 2 employees with same id in different tables , also data stored has redundancy across the tables.

**ALTERNATIVE SOLUTION :** Create a Employee Table which consists of IDs and type of Employee. The ID in the previous three tables should reference the ID of this table as foreign key. This Solution ,however can increase query execution time because of joins in query.

## TRANSACTION DETAILS :

Daily transaction details are stored in **dailywagetrans** table. **DailyTrans** Class is the wrapper for this table. It stores : **emp_id , transaction_date , weekday , amount , type** .

Type is

- '**D**' : for amount earned by employees in a day that work on an hourly basis / or for commission made on sales by a sales employee in a day.
- **'S'** : for time-to-time Service Charges of Union imposed on employees of the union.

**DESIGN PROBLEM 2 :** the size of this table keeps on increasing as data is entered on a daily basis.

**ALTERNATIVE SOLUTION :** clear out data from the table to a data store periodically to keep the size of table under control.

**DESIGN PROBLEM 3 :** method getLastFriday() of DailyTrans class Requires that data is entered for every day for a employee, e.g. we cannot skip daily sales commission for a sales employee on weekends , we have to provide entry with 0 for these days. **This method actually returns the last Friday which appears in a ArrayList<DailyTrans> sequence.**

**DESIGN SOLUTION :** make use of java.util.calendar class (**like in function getLastMonthDay() of SalaryEmployee class** ) to figure out last Friday appearing in a sequence of transactions ordered by date.

## PAYROLL DESIGN MODEL :

For every Employee we **store the last day upto which the payment for a employee is settled**. For

- DailyWageEmployee : this day is a Friday (last_tr_date)
- SalaryEmployee : this is the last working day of a month (last_mtr_date)
- SalesEmployee : we store 2 dates => one is Friday , the other is last working day of month.

A **Payment is said to be settled** till a date 'd' : **All commissions ,weekly wages , salary have been paid till that date & all service charges, weekly dues have been recovered**.

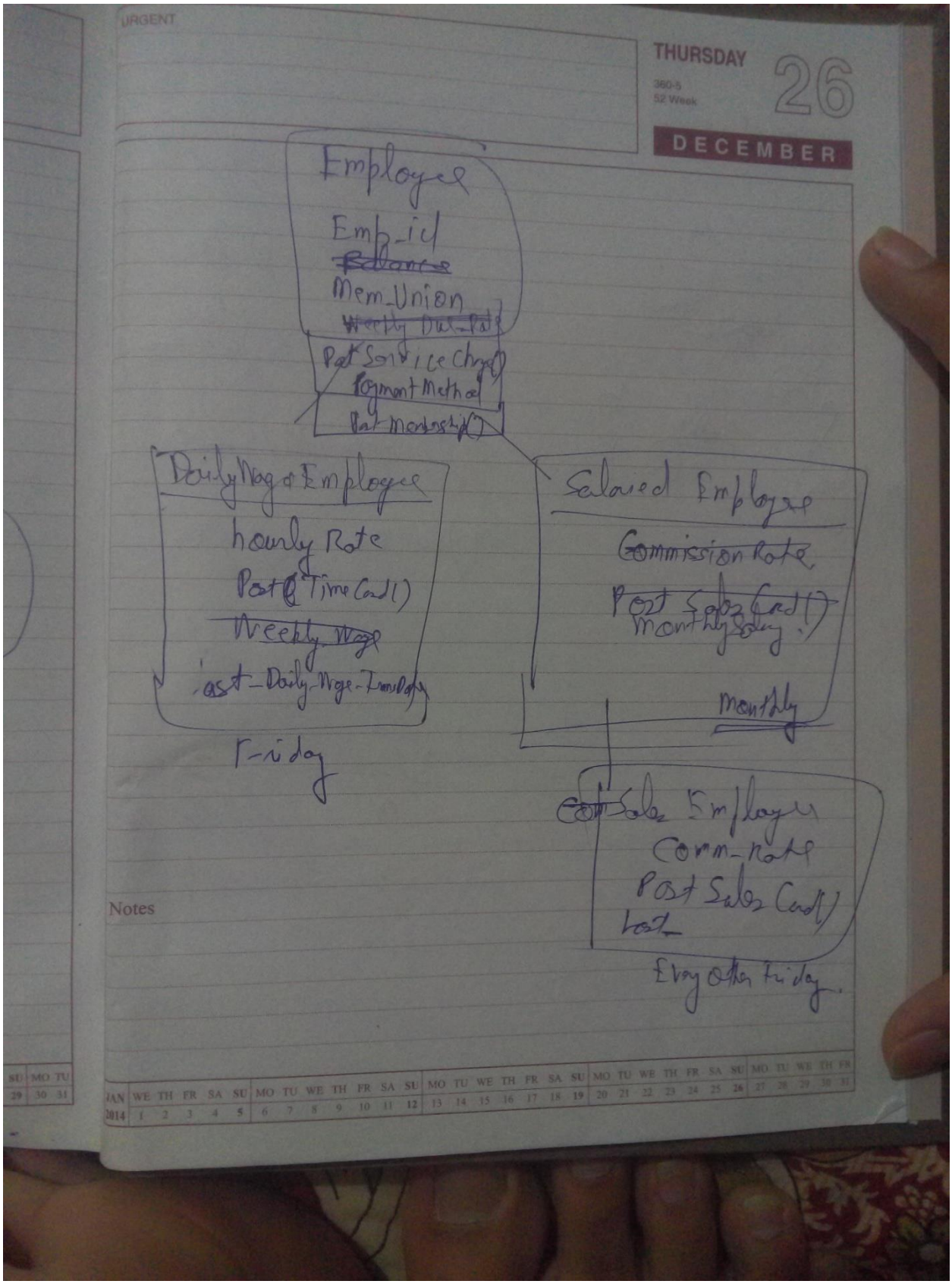
Initially all these dates are null for all employees.

Whenever the payroll is run for an employee , **the payment from last settled date to the closest valid settle date before today** is paid out if the net payout is positive.

**DESIGN ASSUMPTIONS** :

- the transaction table consists data for every day for a employee.(discussed in DESIGN PROBLEM 3)
- every salaried employee is paid his salary for the previous month (i.e. the month before running the Payroll software).
- For Sales Employees , we maintain two dates : the first date stores the date till which all sale payments have been settled , and the second date stores the date till which the salary has been payed.
- For Sales Employees when paying out , the service charges, weekly due rates are deducted till the sales date or salary date ,whichever is later in time.

# ROUGH SHEETS



URGENT

THURSDAY 26
360-5
52 Week
DECEMBER

Employee
Emp_id
~~Balance~~
Mem_Union
~~Weekly Due_Rate~~
Post Service Charge()
Payment Method
Post Membership()

Daily Wage Employee
hourly Rate
Post(Time Card()
~~Weekly Wage~~
Last_Daily_Wage_Transdate

Friday

Salaried Employee
~~Commission Rate~~
~~Post Sales Card()~~
Monthly Salary()

Monthly

~~Com~~ Sales Employee
Comm_Rate
Post Sales Card()
Last_

Every other Friday.

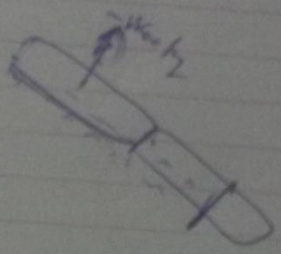
Notes

25 WEDNESDAY

359-6
52 Week

DECEMBER

URGENT



DWE_TRANS

DWE_Transa

Date Type money
DW
SC
WD

D : Daily Wage/Comm.
S : Service charge

A → Addition
P → Pay master
B → B

Notes

| DEC 2013 | SU | MO | TU | WE | TH | FR | SA | SU | MO | TU | WE | TH | FR | SA | SU | MO | TU | WE | TH | FR | SA | SU | MO | TU | WE | TH | FR | SA | SU | MO | TU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | [illegible] | [illegible] | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |