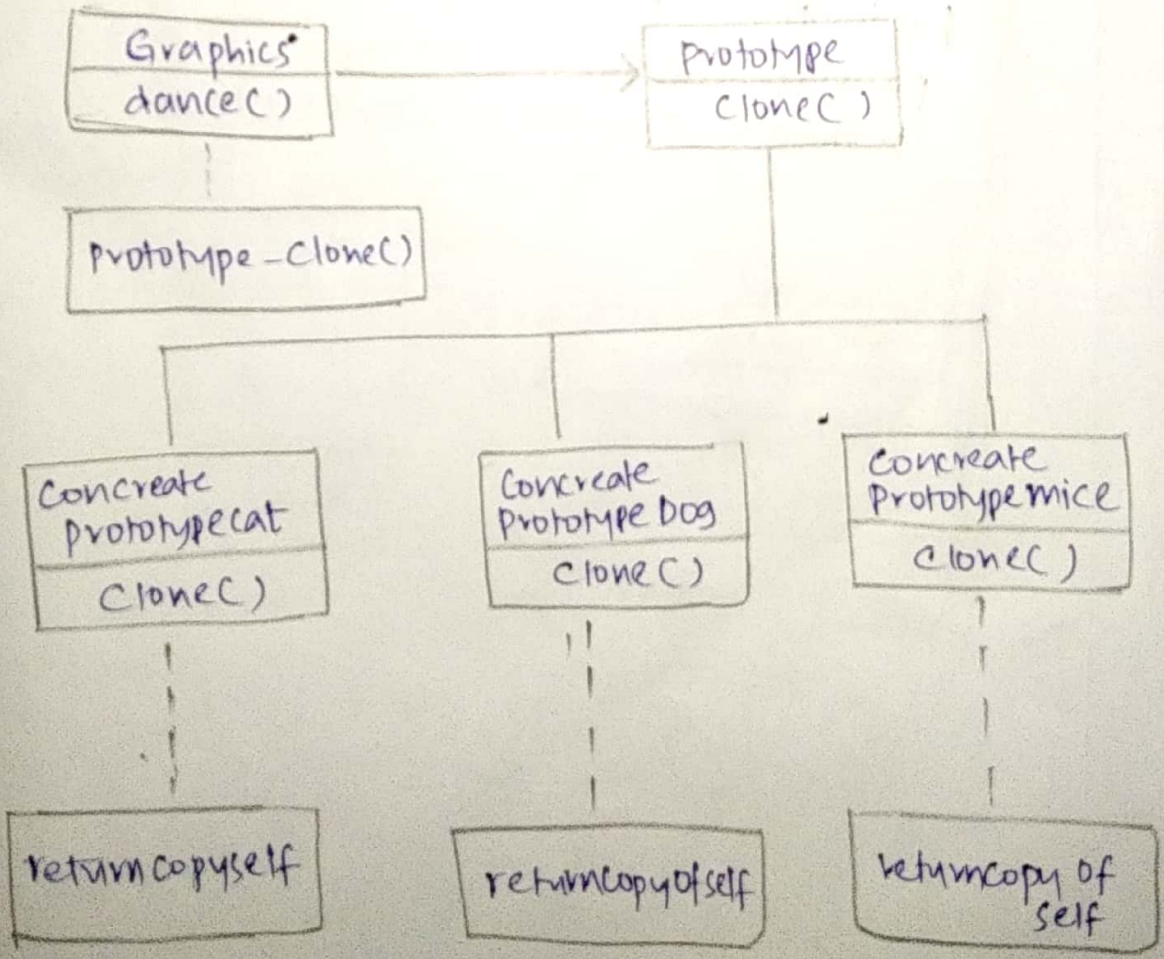
DIGITAL ASSIGNMENT - 3

P. Nityasree - 17MIS1007
P. Reshma - 17MIS1009
B. BLESSY - 17MIS1014

COURSE : SOFTWARE REUSE
COURSE CODE : SWE 2024
FACULTY : Prof Ranganathan Sridhar.

(1) Suggest a design pattern such that you will be able to create an army of cats, dogs, mice they will be able to synchronized dancing. Draw the UML and some sample code for same.

prototype design pattern can be used here, because all are looking similar, but will differ the name only. so we can do cloning



```
Class prototype
{ Friend creatorclass;
private:
   string fvalue;
Public:
   prototype* clone()
    {
    return new prototype(fvalue)}
protected:
    protected prototype ( string avalue):
          { fvalue = avalue;
protected: music ( string dance){
    synchronized ( prototype)
        { prototype(dance);
        } }
```

Test prototype

```
prototype * cat = new prototype ("A value");
prototype * dog = cat. clone();
prototype * mouse = dog. clone();
```

(2.) India – RDBMS

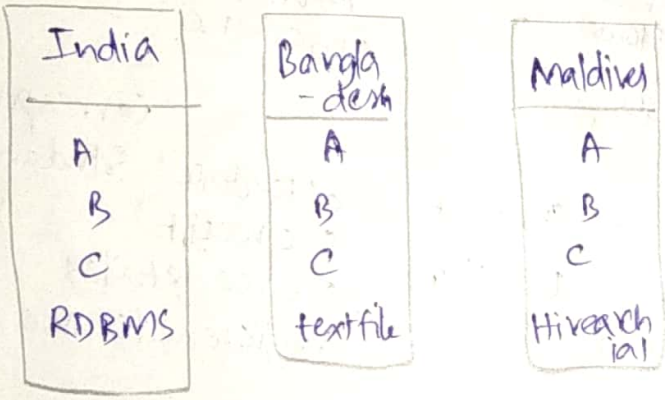
Bangladesh – text file and Arcnet

Maldives – hirearchial database and novel proprietory protocol

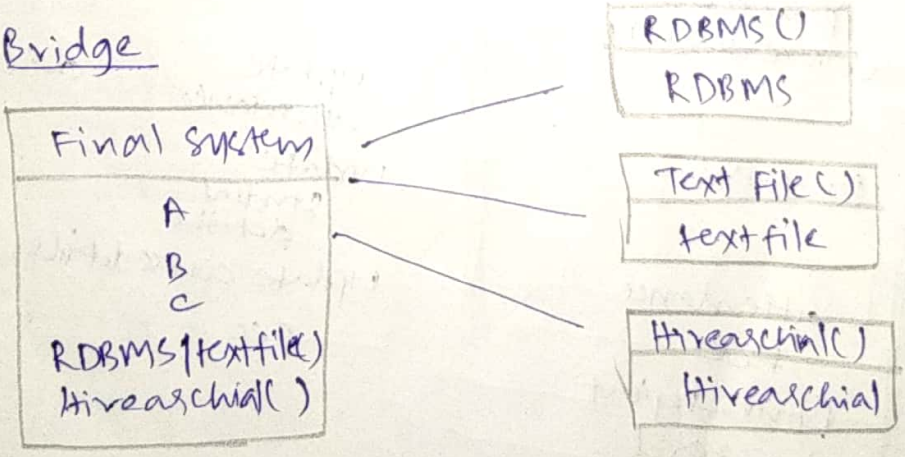
Design pattern : Bridge

It provides alternative implementation of algorithm of system and it is done before system is designed and it is done new system is defined

→ Abstraction and implementation must be used client should be insulated from implementation details.

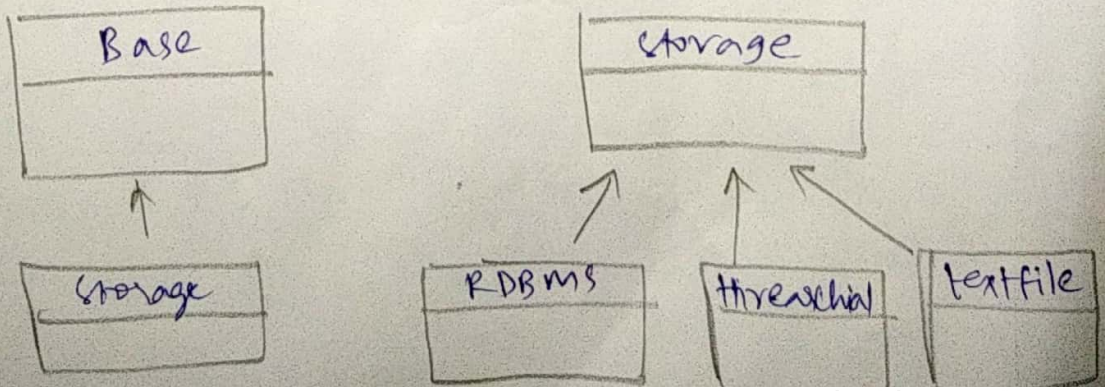


Bridge

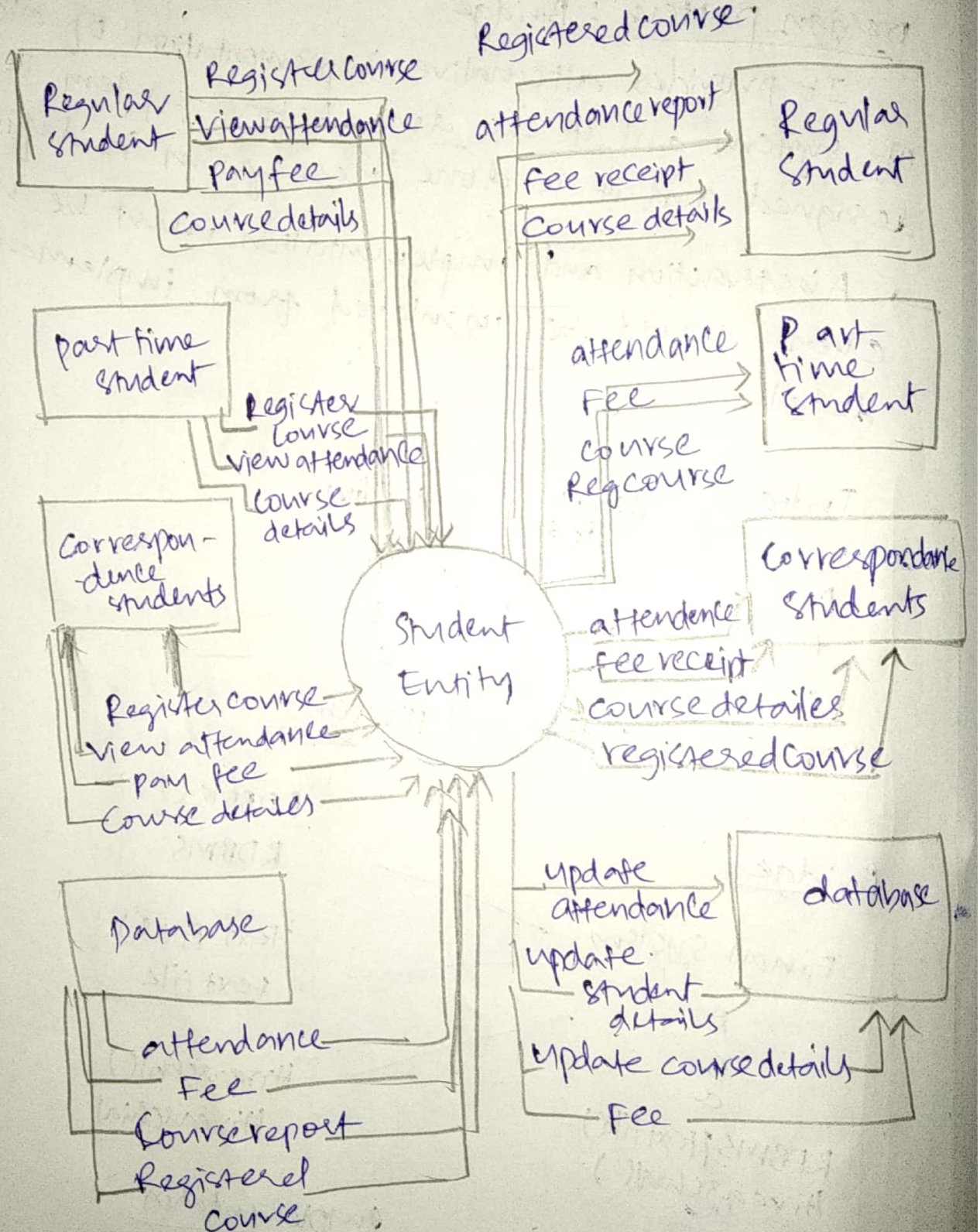


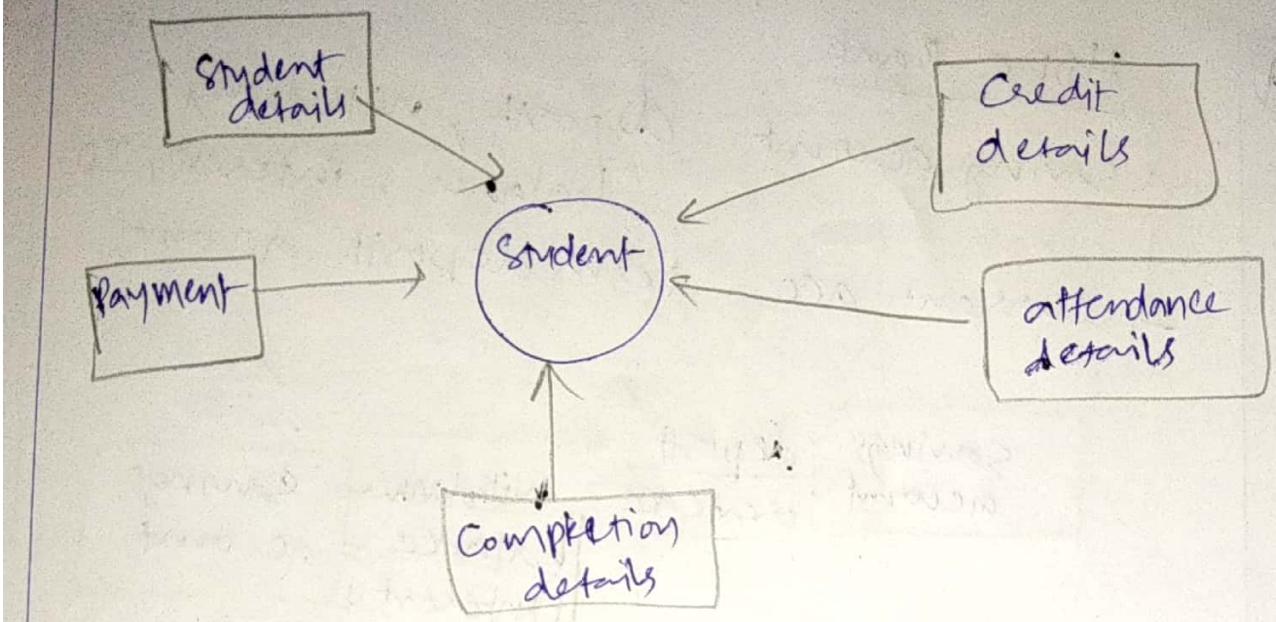
Abstract

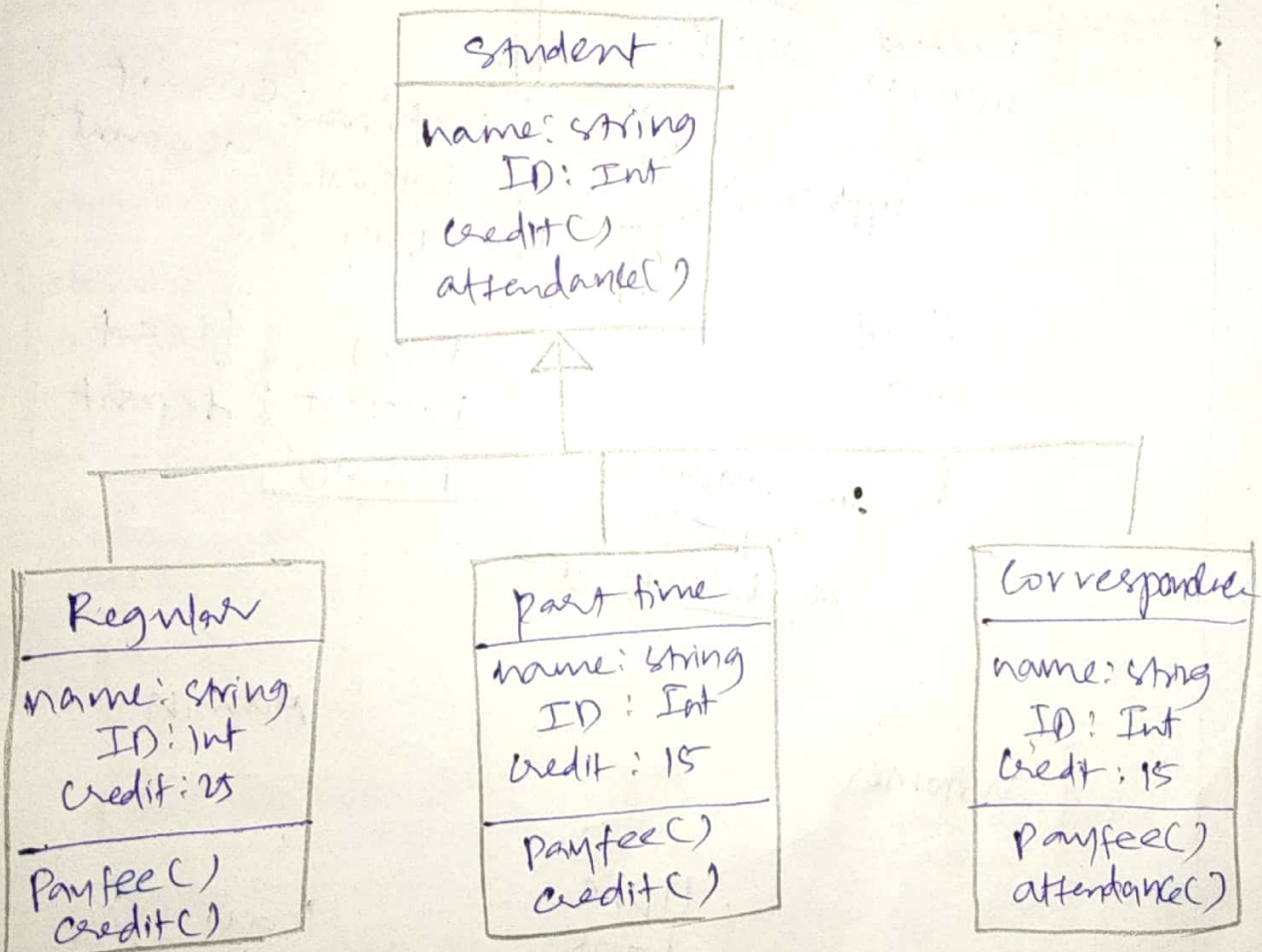
Implementation



(3) Design a OOP system for student entity maximizing software reuse. Draw the UML and explain.





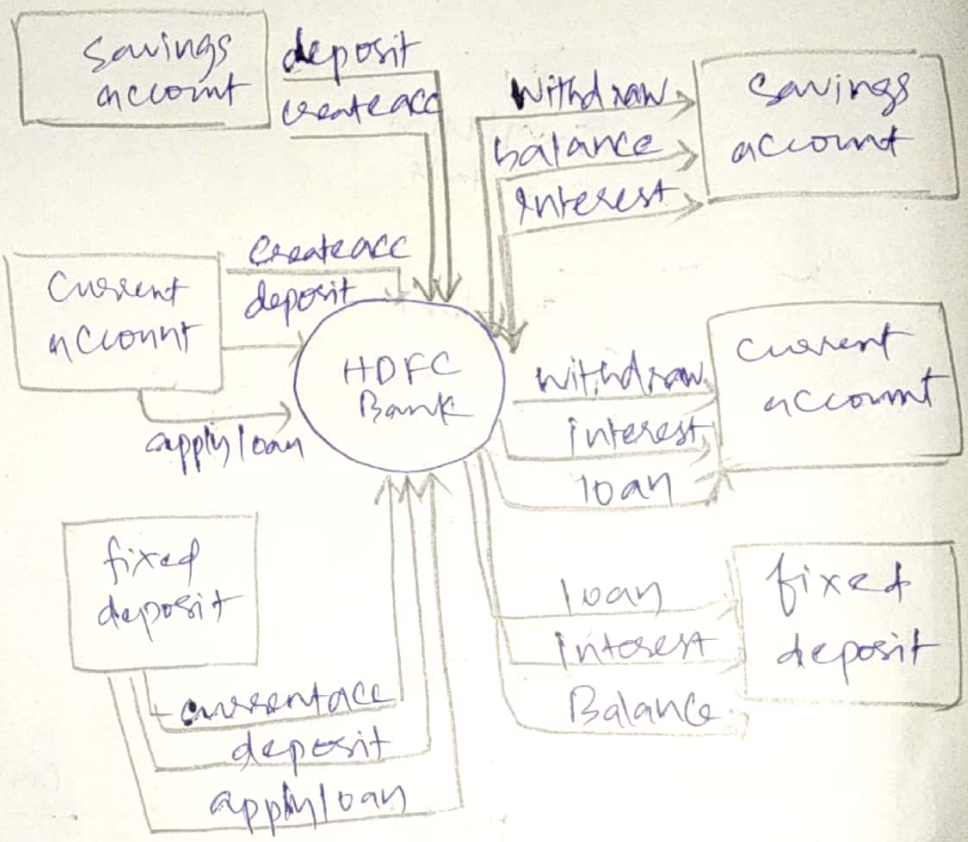


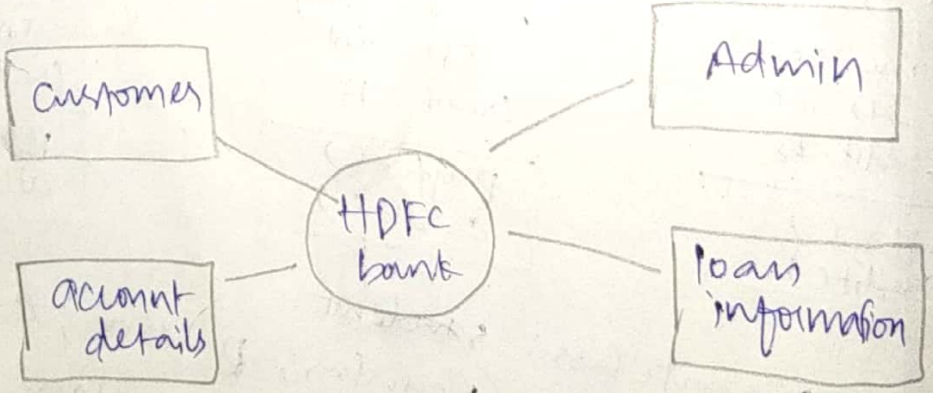
Parent class: student
child class - Regular, parttime, Correspon-dence
Functionalities from the base class will be inherited to class of child.

(4) HDFC Bank

→ saving account - deposit, withdraw, balance, interest, ID

→ Current acc - term deposit acount

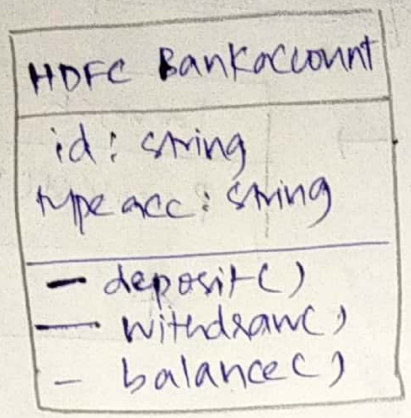


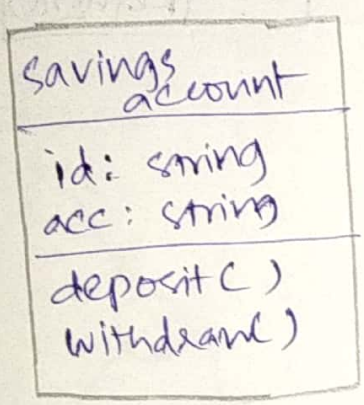


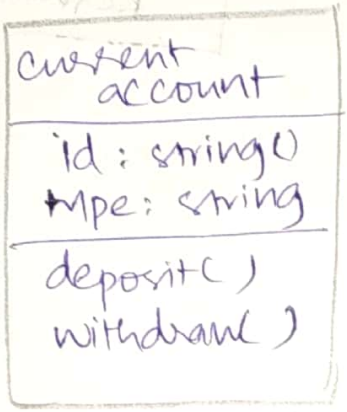
Classes:

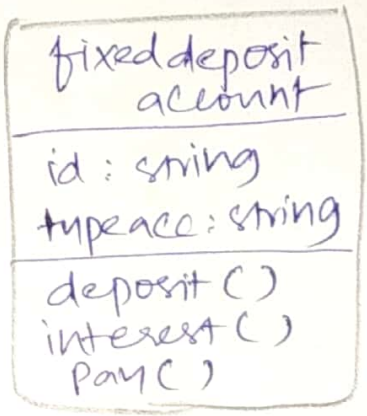
HDFC Bank account is the base class and the savings account, current account and fixed deposit, will inherit the functionalities from the parent class.

class diagram UML):









(5) Legacy system,

User interface wrapping: screens of legacy system captured and mapped to modern graphical interface

function wrapping: not only data but business logic is encoded in old legacy programming lang are wrapped, and accessed wraping, object oriented wrapping.

Design pattern: adapter

Intent: convert interface of class (legacy) into another interface client expect adapter lets classes work together that could not otherwise because of the incompatible interface.

→ Legacy system wrapping can be moving to a new technology as service oriented architecture and the contribution presents a tool supported method for acheiving that goal. legacy code is wrapped which allow individual functions within program to be offered as Webservice to any external users

→ Legacy system a significant part of the company software assets can be pressured in the framework and divided as programs that are dependent, and partially dependent and also completely dependent.

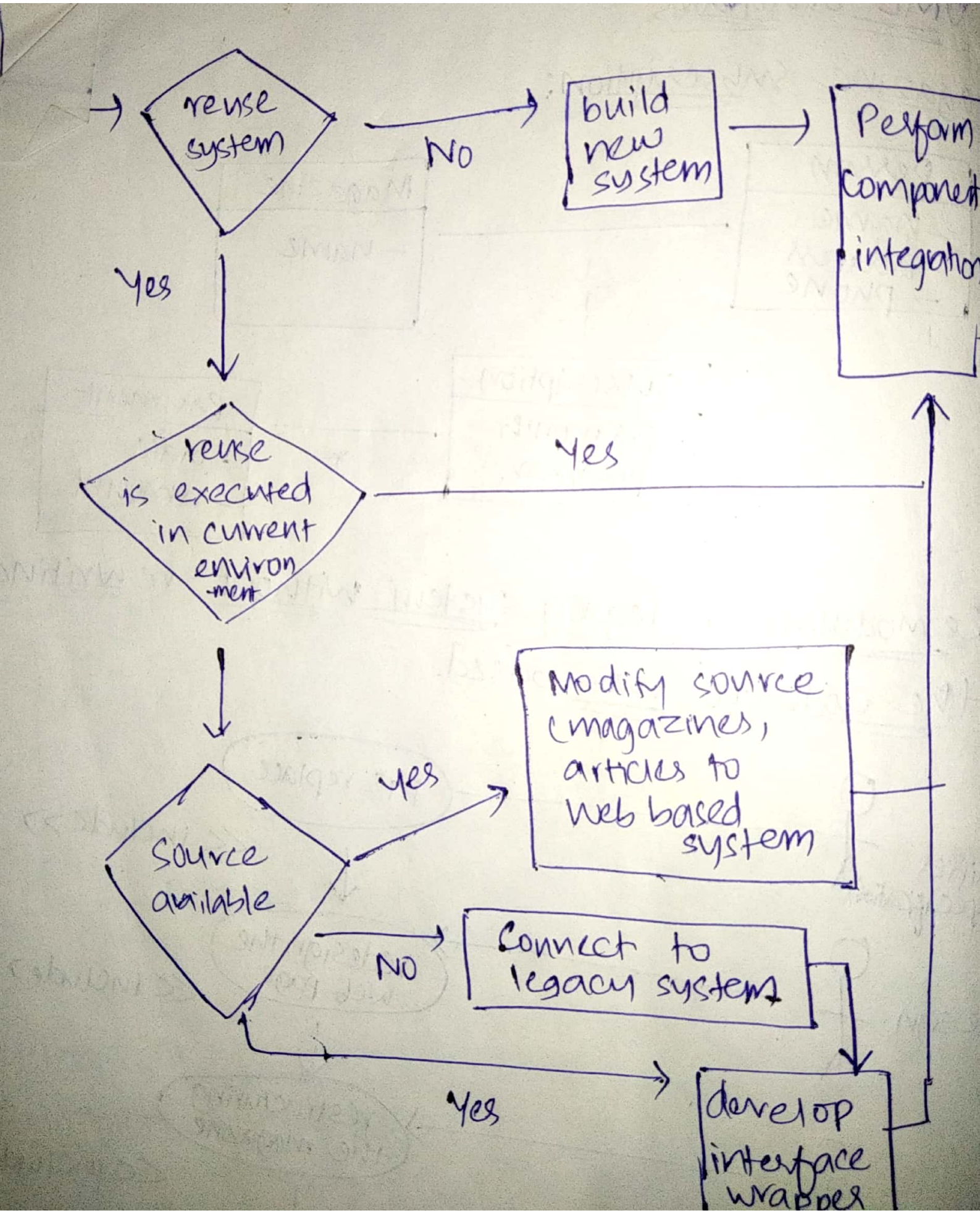
→ The category includes programs written in conventional lang as C, C++ can be reusable in environment.

→ The second category includes programs written in language which uses runtime or link time functions. and third category as natural, CSP, and oracle Frames.

→ The steps to minimize the risk in the legacy management system include as Add the good features which is useful, donot introduce a defect, finish adding features in systematic manner

→ Building a safetynet, adding unit or component test and make room for new feature with refactoring, adding features on legacy code, test manually and refactor clean up.

→ legacy system wrappers will increase the reusability and reduce complexity and expenses.



Structure of class adapter :