waitong suen a1790760 – Hyun Ji Moon a1793295
– OOP Semester 2, 2020 – The University of Adelaide

Project Specification – Major Practical

Introduction

Our project is about library systems such as logging system, borrow-return materials(books, ebooks, DVDs) systems and managing library materials system for staff. These systems allow the users(public, staff) log in, log out, rent materials, return materials, change password. There are more specific works that the library system allows the staff to be able to do such as check history, add materials and delete materials.

Design Description

Assessment Concepts

Memory allocation from stack and the heap

- Arrays: we are using a vector for storing all the materials and a dynamic array for storing the books(DVDs) that the user have borrowed.
- Strings: material names, author names and user names
- Objects : Ebook, book, DVD, public, staff

User Input and Output

- I/O of different data types : Command-line. The users enter

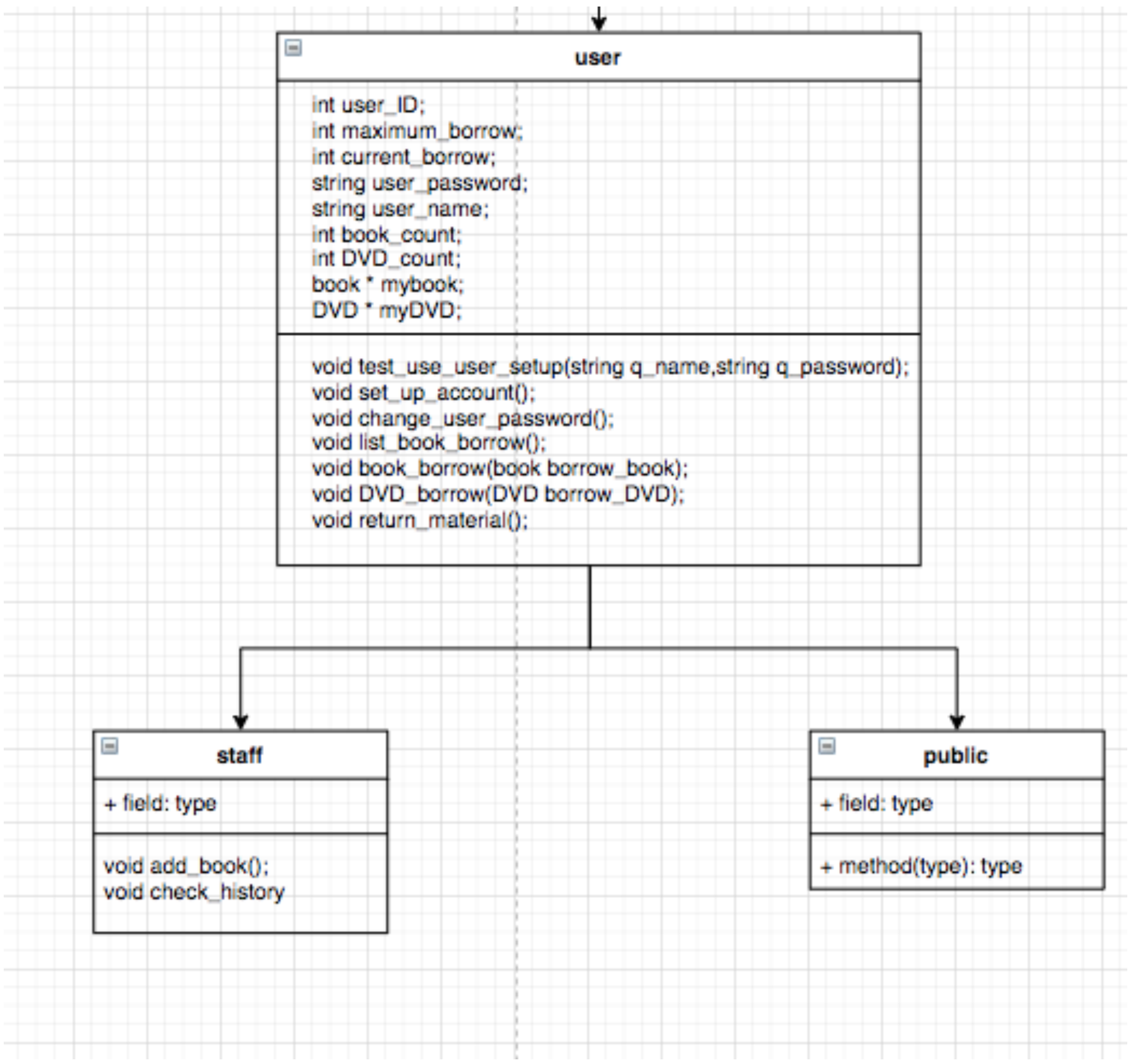
Object-oriented programming and design

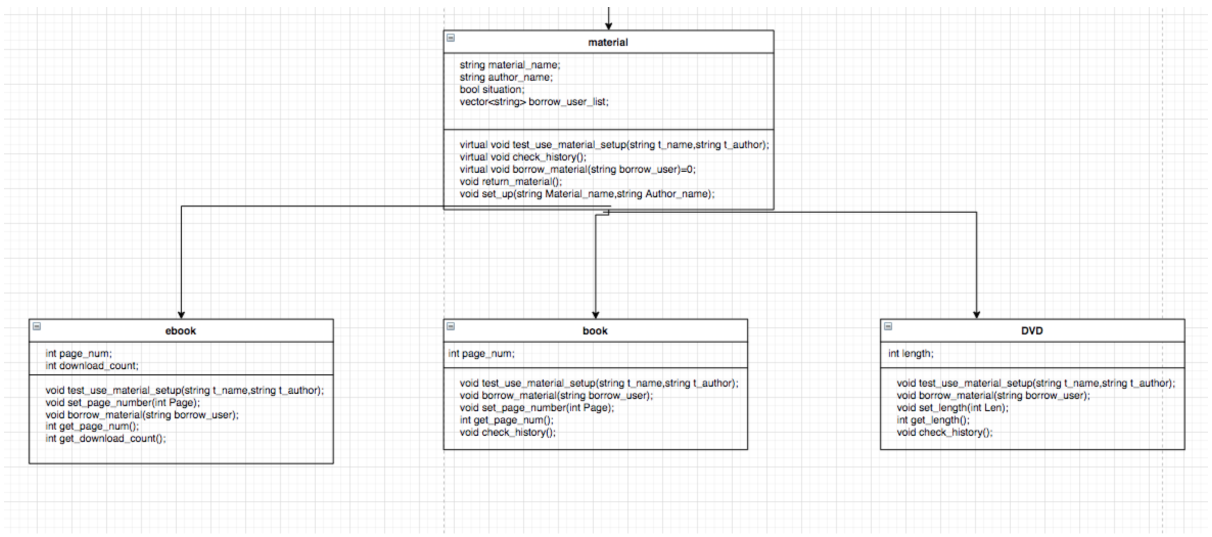
- Inheritance:
  - material(Ebook, book and DVD)
    - There are different types of material such as Ebook, book and DVD.
    - Each child class (Ebook, book, DVD) is inherit from the parent class(material)
    - Each child class has different behaviours to other child class
  - user(staff and public)
    - There are two types of user such as staff and public
    - Both child classes inherit from parent class(user)
    - Each class has different behaviour (staff user can access more library system)
- Polymorphism:
  - test_use_material_setup(string t_name, string t_author)
    : Child classes(Ebook, book and DVD) have the same function for quick setting up the objects to use to debug
  - check_history()
    : Child classes(Ebook, book and DVD) have the same function check_history() but the outcomes are different. The check history for book(DVD) is about the list of the people who borrow the book(DVD)

- Abstract classes:
    - borrow_material(string borrow_user) = 0;

      : every child class has the function for borrowing the material, which has different outcomes like we are going to count out how many ebooks the user has downloaded and the borrowing materials for the book and DVD show the situation of availability of borrowing the book and DVD.

Class Diagram




material
string material_name;
string author_name;
bool situation;
vector<string> borrow_user_list;
virtual void test_use_material_setup(string t_name,string t_author);
virtual void check_history();
virtual void borrow_material(string borrow_user)=0;
void return_material();
void set_up(string Material_name,string Author_name);
ebook
int page_num;
int download_count;
void test_use_material_setup(string t_name,string t_author);
void set_page_number(int Page);
void borrow_material(string borrow_user);
int get_page_num();
int get_download_count();
book
int page_num;
void test_use_material_setup(string t_name,string t_author);
void borrow_material(string borrow_user);
void set_page_number(int Page);
int get_page_num();
void check_history();
DVD
int length;
void test_use_material_setup(string t_name,string t_author);
void borrow_material(string borrow_user);
void set_length(int Len);
int get_length();
void check_history();

Class Descriptions

material:

the superclass of all material in the library.(include book, ebook and DVD), contain attribute:

material name
author_name
situation(of available to borrow or not)
vector<string> borrow_user_list (record the name of user who borrow this material)

ebook:

child class of material, one of the material type. contain extra attribute:

download_count (count the time user download this book instead of borrow)
page_num (pages record)

book:

child class of material, one of the material type. contain extra attribute:

page_num

DVD:

child class of material, one of the material type. contain extra attribute:

length (time length of CD);

user:

the superclass of staff. contain attribute

user_ID;
maximum borrow (limit number of book can borrow)
current borrow (count of recent book borrow)
user_password
user_name
book_count & DVD count (use to calculate the user borrow and check it hit the limit.)
mybook, myDVD   (array for   storing material that user have borrowed)

User Interface

The users (public, staff) is going to input the numbers to choose one of the options using the command-line.

The options that the users can select is like:

What would you like to do today?
1: Borrow material
2: Return material
3. Change current password

4. Check history (only for staff)
5. Add material (only for staff)
6. Delete material (only for staff)

Testing plan

We are going to use our makefile to compile and run our code.

Unit Testing

We have separate test files to check if the individual files are able to be run or not. Each class will be tested individually and through the main function and the main test file, we are going to check if all the classes and codes are running completely.

```
1.
test_add_material :
        g++ add_material_test.cpp  material.cpp add_material_function_list.cpp -o test
2.
test_log_in:
        g++ log_in.cpp user.cpp  material.cpp log_in_test.cpp -o test
```

Schedule Plan

Stretch Goals

Our goal is to make a diagram to check if we satisfy all the rubrics and write down the pseudo code for every class. When we satisfy the requirements, we are going to expand our idea.

week 8	- Write down the diagram - Make a group and creat SVN - Write makefile - make our plan
break	- organise the group works - separate the classes and write some codes - write makefile
week 9	- check if the individual codes are working - write down the main file combining all the codes - write makefile - testing
week 10	- correcting the codes and debug - compile every code
week 11	- check if there is more correction to do - testing