SC/CE/CZ2002: Object-Oriented Design & Programming

ASSIGNMENT

Building an OO Application

2022/2023 SEMESTER 2

SCHOOL OF COMPUTER SCIENCE & ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

1. OBJECTIVE

The main objective of this assignment is

- to apply the Object-Oriented (OO) concepts you have learnt in the course,
- to model, design and develop an OO application.
- to gain familiarity with using Java as an object oriented programminglanguage.
- to work collaboratively as a group to achieve a common goal.

2. LABORATORY

Assigned SCSE lab.

3. EQUIPMENT

Hardware: PC (or Laptop)
Software: Your prefered Java IDE or simply notepad and Java Development ToolKits(JDK)

4. THE ASSIGNMENT

The assignment for your group will be to design and develop a:

Final Year Project Management System (FYPMS).

FYPMS is an application for staffs and students to manage FYP.

The following are information about the application:

Study the following description of a simplified Final Year Project (FYP) system and implement **ALL** the functions.

About User

a) The users can login to the FYP system using userID and password.
b) The users include supervisors, students, and FYP coordinator, who is a supervisor as well.

About userID and password

a) The userID is the NTU network user ID, that is the part before @ in email address.
b) Assume all users use the default password, which is password.
c) A user can change password in the system.

About Student

a) Student list is uploaded using a file when system is initialized.
b) In initialization file, each record contains Name and email address.
c) StudentID is the userID of a student.
d) A student can view all available projects if he/she has not registered a project.

e) A student can select a project and request the FYP coordinator to allocate the project to him/her, after that, once the FYP coordinator approves the request, the registration is done.
f) A student only can view his/her registered project once the registration is done.
g) A student can **request** the supervisor to change title to a student proposed one after the registration is done. Once the supervisor approves the request, the title will be changed.
h) A student can **request** the FYP coordinator to deregister FYP after allocation. Once the FYP coordinator approves the request, it is deregistered. Please note that the student is not allowed to select project again later after doing so.
i) A student can view his **requests'** history and status.

About Supervisor

a) Faculty list is uploaded using a file when system is initialized.
b) A supervisor must be a faculty.
c) supervsiorID is the userID of a faculty.
d) A supervisor can create many projects, one at a time.
e) A supervisor can view the information of all the projects submitted by him/her.
f) **Only** the supervisor can modify the title of a project submitted by him/her.
g) A supervisor can modify the title of all the projects submitted by him/her.
h) A supervisor can modify the title of a project by approving student's request.
i) A supervisor cannot supervise more than 2 projects. Once the cap is reached, all his/her remaining available projects will become unavailable.
j) A supervisor can **request** FYP coordinator to transfer a student to a replacement supervisor by providing project ID and replacement supervisorID. Once the FYP coordinator approve the request, the student will be transferred to the replacement supervisor.
k) A supervisor can view all the pending **requests** from students and approve/reject them.
l) A supervisor can view his incoming and outgoing **requests'** history and status.

About Request

a) Request from student to supervisor
b) Request from student to FYP coordinator
c) Request from supervisor to FYP coordinator

About Project

a) FYP is an individual work.
b) Available Project information displays projectID, supervisor Name, supervisor Email address, projectTitle, and status.
c) Project status is either available, reserved, unavailable or allocated.
d) Allocated Project information displays projectID, supervisor Name, supervisor Email address, student Name, student Email address, projectTitle, and status.

About Project creation

a) When the system runs at first time, the rollover project file is loaded to initialize the project list.
b) A project can be manually created by a supervisor.
c) Once a supervisor submits a new project to the FYP system, the supervisorID will be the faculty's userID and projectID will be an automatically generated sequential index, the default status is available and studentId is empty.

About Project selection(It is different from the real SCSE FYP system, for assignment only):

a) First come first serve

b) The student can view the available project list and choose one, the status of the project will be changed to reserved and will be excluded from available project list. Once the **request** is approved. the project chosen will be allocated to the student. The studentID of the project will be updated to be the studentID and status will be updated to be allocated. If the request is rejected, the status of the project will be changed back to available.

c) Assume that there are more available projects than the number of students.

About Project deregistration:

The deregistered project will be recycled and add back to available project list.

About FYP coordinator :

a) FYP coordinator information is uploaded using a file when system is initialized.

b) **Only** the FYP coordinator can change supervisor of a project upon request.

c) **Only** the FYP coordinator can allocate a project to a student upon request.

d) Upon request, **only** the FYP coordinator can deregister a student from FYP by changing the studentID of the allocated project to be empty and project status back to available, at the same time, update the student to be deregistered.

e) FYP coordinator can view all available, unavailable, reserved and allocated projects.

f) FYP coordinator can generate project details report according to searching filters, e.g., status, or supervisor.

g) FYP coordinator can view all pending requests sent by supervisors and students and approve/reject them.

h) FYP coordinator can view all requests' history and status.

The application is to be developed as a **Command Line Interface (CLI) application (non-Graphical User Interface).**

The sample data files student list, project list, faculty list and FYP coordinator list are given in excel in assignment folder. You can

- **use them directly,**
- **or copy the content to text file if you will read from text file,**
- **or make you own data files.**

But ***No database application (eg MySQL, MS Access, etc) is to be used.***
No JSON or XML is to be used.

5. THE REPORT

Your report will include the following :

a) A detailed UML **Class** Diagram for the application (exported as an image)
   - show clearly the class relationship, notation
   - notes to explain, if necessary

b) A **write-up** on your **design considerations** and use of OO concepts in your current design.
   - , extensibility and maintainability of your design.

c) Reflection: The difficulties encountered and the way to conquer, the knowledge learnt from this course, further improvement suggestion. Strong demonstration of learning points and insights of good design and implementation practices, based on experience gained from doing the assignment.

d) A duly signed **Declaration of Original Work** form (Appendix B).

e) [**Optional**] Member's work contribution and distribution breakdown. *If your group feels that marks should be given based on contribution, your group can fill up the WBS.xls(in the same folder as assignment doc) and include it in this report. All members MUST consent to the WBS contents.* ***You must also email the WBS.xls to the course-coordinator with ALL members in the loop.***

6. DEMONSTRATION

Your group is to produce a **video and audio recording** to demonstrate the working of the application – presenting ALL the required functionalities of the application. It is advised that you planned your demonstration in a story-boarding flow to facilitate understanding of your application. *Please introduce your members and group number at the start of video, all the group members must take turn to present. The presenter should show his/her face while presenting.*

In the production, you may include:

a) Explaining essential and relevant information about the application

b) Run-through and elaborate on essential part/s of your implementation/coding

- ***The video duration must not exceed 15 minutes in total.***
- ***The font size used must be large enough to be readable and viewable.***
- ***The video quality must be clear.***
- ***The demo of the application is to done in real-time and NOT pre-run display.***

You will create your own test cases and data to test your application thoroughly. *Refer to **Appendix A for reference.*

7. THE DELIVERABLE

Your group submission should include the following:

a. The report (separate diagram file if diagram is unclear in report)

b. Video and audio recording of the demonstration.

c. All implementation codes and java documentation (javadoc).

d. Other relevant files (eg data files, setup instruction, etc)

8. ASSESSMENT WEIGHTAGE

UML Class Diagram [**30 Marks**]

- Show mainly the Entity classes, the essential Control and Boundary classes, andenumeration type (if there is).
- Clarity, Correctness and Completeness of details and relationship.

Design Consideration [**15 Marks**]

- Usage of OO concepts and principle - correctness and appropriateness

Implementation Code [**35 Marks**]

- Diagram to Code correctness, readability, Javanaming convention, exceptionhandling, completeness of Java Doc and overall quality.
- A Java API HTML documentation of **ALL** your defined classes using Javadoc mustbe submitted. The use of javadoc feature is documented in Appendix D.

Demonstration and report [**20 Marks**]

- Coverage of application essentials and functionalities, user friendliness, demoflow, innovation.
- Report structure and reflection

9. SUBMISSION

This is a **group assignment**, and one submission from each group.

Report format guidelines are provided in the Appendix C below.

1. Soft copy of your deliverables to be **uploaded** to your individual CE/CZ2002 **LAB site** (eg FEP1, FSP1, etc) in **NTULearn**. The link is provided on the left panel "Assignment Submission".
   File name convention : <lab_grp>-grp<assignment_grp#>.<ext> Eg, FEP2-grp3.pdf [<ext> can be pdf, doc, zip, mpeg, wmv or mp4]
   [In the event your video file is too large to upload, you may upload to **Youtube** and **provide the link in your report**. **No online storage** (like DropBox, Google Drive, etc) is allowed – no updates after submission].

2. **DEADLINE** : **Week 13 Sunday, 11.59pm.**

Important:

Note that THREE (3) marks will be deducted for the delay submission of each calendar day. Lateness is based on the date the captured in NTULearn or subsequent resubmissions (whichever is later). So check your work before submitting.

-

10. **REFERENCES & TOOLS**

- UML Diagrams tool - Visual Paradigm http://www.visual-paradigm.com/
- http://www.visual-paradigm.com/support/documents/vpuserguide/94/2576/7190_drawingclass.html
- NTULearn Cx2002 main course site content
- NTULearn Cx2002 course site content on "File Input/Output"
- Object Serialization tutorial http://www.javabeginner.com/uncategorized/java-serialization
- Windows Media Encoder (a suggestion) http://www.microsoft.com/expression/products/EncoderPro_Overview.aspx
  [You can also try with the video recording feature for gaming in Windows 10 – press 'Windows key + G']

APPENDIX A:

Suggested Test cases

The list of sample test cases are guide for your testing and demo video.
Depending on your design and user-friendliness of your data entries process,
there may be multiple steps taken.
[Note : You should also demonstrated at least 5 cases of input error checking done in yourapplication]

a. Login
- Cannot login: invalid user or valid ID but wrong password
- After login successfully, different menu list (main page) displayed for different user category, **faculty**, **students,** and **FYP coordinator**.
- After login, the user is able to change password. [expected: re-login to verify the effect]

b. The main page for students should have functions after login as follows (the detail should strictly follow the above description in Section 4):
- Change password.
- View available projects.
- Select the project to send to the coordinator.
- View his/her own project.
- View requests status and history.
- Request to change the title. Input the new title and the supervisor should get the request.
- Request coordinator to deregister FYP.

c. The main page for supervisors should have functions after login as follows (the detail should strictly follow the above description in Section 4):
- Change password.
- Create/update/View projects:
- Input choice from ‘Create’, ‘update’, ’view’ to choose function
- Create a project by inputting the required data.
- ‘View’ will print the necessary data.
- When choosing ‘Update’:
    - Choose a project by projectID.
    - Input a new title to the update the old one of a project.
- View student pending requests (indicate *NEW* if there is pending request):
    - Choose student request by studentID.
- Approve request.
- Reject request.
- View request history.
- Update the title of project by approving the request.
- Request to transfer student by inputting necessary input and the FYP coordinator should get the request.

d. The main page for coordinators should have all the functions after login as follows (the detail should strictly follow the above description in in Section 4):

- **All the functions that supervisor main page have**
- View request (indicate *NEW* if there is pending request):
  - Approve request.
  - Reject request.
  - Choose the request from supervisors and change.
    - Transfer student
    - If the replacement supervisor has already hit the cap, the hint message should be displayed and up to the coordinator's decision to approve or reject the request
  - Allocate a project to students as their requests:
    - View requests.
    - Choose requests and allocate as requests.
  - Deallocate projects from students:
    - View requests.
    - Choose requests and deallocate as requests.
- View projects according to different filters.

Sample scenario:

Student #1

- View registered project fails. Error message: *You have not registered a project.*
- View all available projects
- Select a project by specifying the projectID. Request for registration.
- …*FYP coordinator approves request*
- View all available projects fails. Error message: *You are currently allocated to a FYP and do not have access to available project list.*
- View registered project.
- Request to change title by providing a new title.
- … *Supervisor approves request*
- View registered project to verify the title change.
- Request to deregister the project.
- … *FYP coordinator approves request*
- View registered project fails. Error message: *You have not registered a project.*
- View all available projects fails. Error message: *You are not allowed to make selection again as you deregistered your FYP.*

Supervisor's cap

- Student #2 selects Bo An's project
- Student #2 changes title
- Student #3 selects Bo An's project
- Student #4 view all available projects, Bo An's remaining projects are NOT included in the list.
- Bo An transfers a project to Dusit Niyato.
- Student #4 view all available projects, Bo An's remaining projects are included in the list.
- Student #4 selects Dusit Niyato's project.
- Dusit Niyato submits a new project.
- Student #5 view all available projects, Dusit Niyato's remaining projects are NOT

included in the list.

- Student #4 deregisters FYP.
- Student #5 view all available projects, Dusit Niyato's remaining projects including the deregistered project will be displayed in the available project list.
- Student #5 selects the recycled project.

FYP coordinator

- Submit a project.
- *…A student selects this project.*
- Approve the request.
- *…The student requests to change title.*
- Approve the request.

Note: Need show in class diagram and code that

a) **Only** the supervisor can modify the title of a project submitted by him/her.
b) **Only** the FYP coordinator can change supervisor of a project.
c) **Only** the FYP coordinator can allocate a project to a student.
d) **Only** the FYP coordinator can deregister a student from FYP.

APPENDIX B:

Declaration of Original Work for CE/CZ2002 Assignment

We hereby declare that the attached group assignment has been researched, undertaken, completed andsubmitted as a collective effort by the group members listed below.

We have honored the principles of academic integrity and have upheld Student Code of AcademicConduct in the completion of this work.

We understand that if plagiarism is found in the assignment, then lower marks or no marks will beawarded for the assessed work. In addition, disciplinary actions may be taken.

Name	Course (CE2002 or CZ2002)	Lab Group	Signature /Date

Important notes:

1. Name must **EXACTLY MATCH** the one printed on your Matriculation Card.

APPENDIX C:

Report requirement:

1. **Format**:

For the main content, please use Times New Roman 12 pt font size and 1.5 linespacing. You may choose to use other fonts (e.g, Courier New) for code segments. Please use the following report structure:

- Cover page: Declaration of original work (Appendix B)
- Design Considerations .
  - Approach taken, Principles used, Assumptions made, etc
  - ***Optional*** : You can show the important code segment (e.g, a method or a few lines of code) and necessary illustrations toexplain your solution.
- Detailed UML Class Diagram.
  - Further Notes, if needed
- Testing.
  - Test Cases and Results
- Reflection.
  - The difficulties encountered and the way to conquer, theknowledge learnt from this course, further improvement suggestion.

2. **Length**:

The report should be at most 12 pages from cover to cover including diagrams/Testing results/references/appendix, if there is any. If you could wellpresent your work in fewer than 12 pages, you are encouraged to do so.

DO NOT include source code in the report but stored the source code in a folder. You are to ensure that the diagrams are readable and clear to the reader. [You can save the diagrams as image files and include in a folder]

APPENDIX D:

Creating Javadoc:

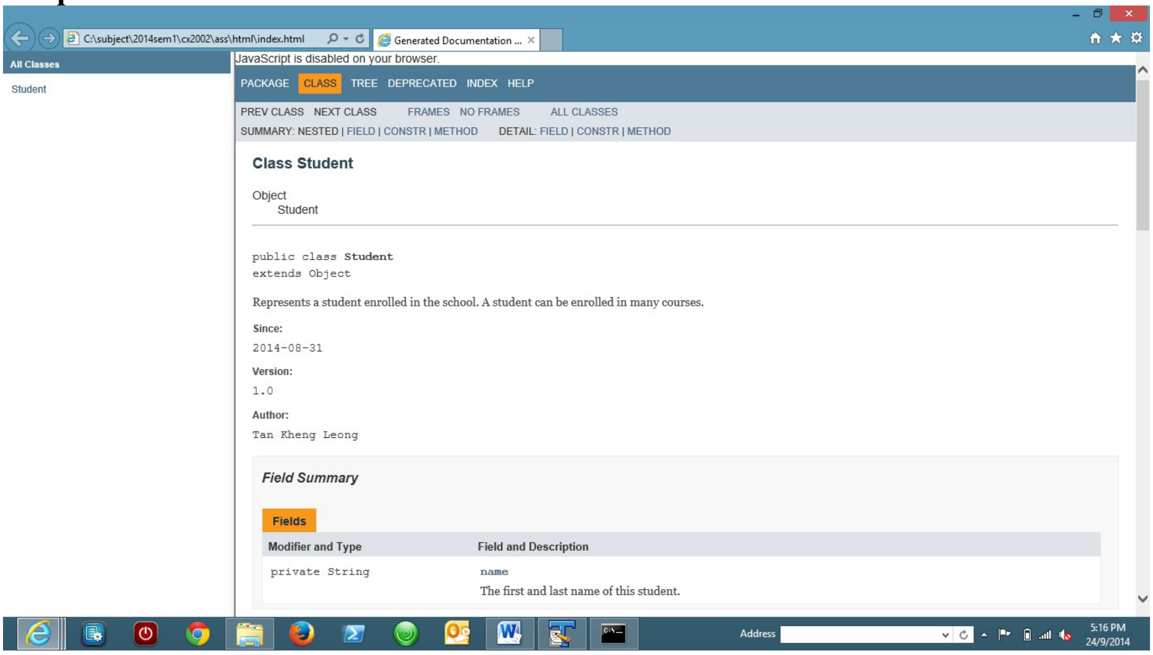
Detailed can be found at
http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html

Using Javadoc in Eclipse : Youtube : http://www.youtube.com/watch?v=Hx-8BD_Osdw

Below is a short example :

```
/**
  Represents a student enrolled in the school.
  A student can be enrolled in many courses.
  @author  Tan Kheng Leong
  @version 1.0
  @since   2014-08-31
 */
public class Student {

  /**
   * The first and last name of this student.
   */
  private String name;

  /**
   * The age of this student.
   */
  private int age;

  /**
   * Creates a new Student with the given name.
   * The name should include both first and
   * last name.
   * @param name This Student's name.
   * @param age  This Student's age.
   */
  public Student(String name, int age) {
    this.name = name;
    this.age = age;
  }

  /**
   * Gets the first and last name of this Student.
   * @return this Student's name.
   */
  public String getName() {
    return name;
  }

  /**
   * Changes the name of this Student.
   * This may involve a lengthy legal process.
   * @param newName This Student's new name.
   *                Should include both first
   *                and last name.
   */
  public void setName(String newName) {
    name = newName;
    }

}
```

Output from Javadoc – index.html


C:\subject\2014sem1\cx2002\ass\html\index.html
Generated Documentation ...
All Classes
Student
PACKAGE CLASS TREE DEPRECATED INDEX HELP
PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD
Address
5:16 PM
24/9/2014

JavaScript is disabled on your browser.

Class Student

Object
    Student

```
public class Student
extends Object
```

Represents a student enrolled in the school. A student can be enrolled in many courses.

Since:
2014-08-31

Version:
1.0

Author:
Tan Kheng Leong

Field Summary

Fields

Modifier and Type	Field and Description
private String	name The first and last name of this student.

<u>**For those familiar with using command prompt :**</u>

Steps to general API doc :

(1) Locate the installed path of JDK (java development kit)
  - In Windows, it should be in C:\Program Files\Java\jdk<version>\
(2) Open command prompt
(3) Go to your src directory using cd
(4) At prompt …..src> <path to jdk>\bin\javadoc" -d ./html -author -private -noqualifier all -version <packagename1> <packagename2> <….>

Eg .
C:\subject\2014sem1\cx2002\src>"C:\Program Files (x86)\Java\jdk1.8.0_05\bin\javadoc" -d ./html -author
-private -noqualifier all -version edu.ntu.sce.cx2002 edu.ntu.sce.cx2003

Statement	Purpose
C:\subject\2014sem1\cx2002\src>	Path to your src root
"C:\Program Files (x86)\Java\jdk1.8.0_05\bin\javadoc"	Path to your jdk javadoc.exe [using double quote if path has space in between, eg Program Files]
-d ./html	-d : specific folder to store html doc Eg ./html means current directory create a html folderto store
-author	Include @author in doc, if provided
-private	Include all methods and fields
--noqualifier all	Omitted all full package name. Eg show **String** instead of **java.lang.String**
-version	Include @version in doc, if provided
edu.ntu.sce.cx2002 edu.ntu.sce.cx2003	Different package names