CMPE133-01 Lab6 Project Demo

Positive Coders

Yunfan Wang <yunfan.wang@sjsu.edu>

Karma Pandya <kay.pandya@gmail.com>

Yehia Qtaish <Yehiaqtaish@hotmail.com>

# 1- Class Diagram

# EBTs and BOs

In this team project, we have three EBTs: Compliance, Fulfillment and Support.

The BOs are AnyForm, AnyInterface, AnyTaskm, AnyImpact, AnyCategory and AnyAspect.

# Functional Requirement

According to these EBTs and BOs, we create the following functional requirement.



Figure : functional Requirement

# Scenario Description

This scenario is about university courses searching engine website. Students can search course records in this website. After clicking the searching results, which are the wanted courses, the students can enroll in these courses.

The following diagram illustrate the application.



Figure : Application

The following are several steps of this scenario:

1. AnyForm(Schedule) decides AnyInterface(webInterFace) for user
2. AnyInterFace(WebInterface) follows AnyRule(SpecificRule)
3. AnyTask(Searching) makes AnyImpact(enrolling)
4. AnyImpact(enrolling) records AnyLog(SearchingHistory)
5. AnyLog(SearchingHistory) is on AnyMedia(Internet)
6. AnyTask(searching) have AnyRule(SpecificRule)
7. AnyType(SpecificType) has AnyEntity(Course)
8. AnyEntity(Course) is stored on AnyMedia(Internet)
9. AnyCategory(Department) varies AnyAspect(SpecificAspect

2- Assumed Factors: It could depend on the amount of times a user prompts a search. Could affect the time it takes.

3- Teacher, Student**.java**: Both play role of AnyParty. Exchange information between two parties. Course, department, schedule, teacher.**java:** has different information regarding Entity, Type, Rule.

4- Class: operations and attributes:

**a)** AnyEntity IO: **attributes**: name,id,type. Related to course where course gets operations from.  
**Operations:** setId(), getID(), setName(),getName(), getType(), setType().//Different set,get methods to get the information regarding course. toString()//Method to print out all information.  
**b)** Course BO: **attributes**: name, time, id, department// Fields for course information  
getName(),getNumber(),getTime(),getDerpartment().//Operations to get information regarding course. toString(): print out course information. related to AnyEntity

**c)** AnyForm BO: **attributes:** ArrayList<Object>//Store schedule.

**operations:** setArr(),getArr().//Methods to form schedule array.

**d)** Schedule IO: **Operations:** addCourse() // adds courses into AnyForm array.  
print()//prints out the array. to AnyForm  
**e)** AnyImpact BO: **attributes:** ArrayList<Object> // store enrolling elements.  
**operations:** setArr(),getArr().//Methods to form enrolling array.

**f)** Enrolling IO: **operations:** addCourse()//adds course to array in AnyImpact. Related to AnyImpact

**g)** AnyLog BO: **attributes:** Arraylist to store search history.  
**operations:** addElementLog()//adds element to log array. printLog()//prints log array.

**h)** searchHistory IO: **operations:** addHistory()//adds searched elements in array  
printHistory()//prints Impact Array.related to AnyLog

**i)** AnyParty BO: **attributes:** userName, name,id,units,password.

**operations:** getuserName(),getNumber(),getTime(),getDerpartment(). getName(), setName(),setUnits(),setid(),setPassword()//methods to intialize party information.

**j)** Studen IOt: **attributes:** maxUnits, studentUnits, enroll, rule, schedule.//last 3 are other classes that is connected to it. related to AnyParty

**operations:** getPsd(),getEnroll(), getRule(), getMaxUnits(), getStudentUnits(), setStudentUnits(),//Gets information from AnyParty to set student main information. checkUnits()// make sure student doesnt break Rule.

**k)** Teacher IO: information could be set from AnyParty. Related to AnyParty

**l)** AnyRule BO: **Operations:** check(arr) check if is true or not.   
**m)** EnrollingRule IO: **operations:** check()//makes sure doesnt pass the maxUnits. related to ANyRule

**n)** LuceneTest: tests lucene, **operations:** addDoc()// adds document to lucene.

**o)** MyServlet: **attributes :** w //index writer config// indexwriteconfig, index//dictionary,log/searchhistory std//students tch//teachers, currentStudent//Student, courseList//course

**operations:** processRequest(): connects to server. addDoc()//adds document. goToSearch(), gotolog(), differnt actions when opening website. Goes to search or log file. gotomessage() displays message after logging in. gotoEnroll() goes to enroll page. goaddCourse() goes to page to add courses;