Architecture Design

for

Graduation Placement Service (GPS)

Version 1.0 approved

Prepared by Joshua(Full), Vicky(Full), Knight(Full), Frye(Some), Unknown(Some)

Cherry

2023/4/4

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Joshua	2023.4.4	First Draft	1.0

1. Overview

1.1 Project description

The purpose of the Graduation Placement Service (GPS) project is to help UIC undergraduate students know more information about admissions of postgraduate (PG) universities and jobs through the data of previous UIC graduates, so that students can have a better understanding of their own situation and make appropriate choices.

1.2 References

SRS_v1.3.pdf
Graduation Placement Service v3.docx

1.3 Design purpose

To make developer easily design this software. To convert the complicated description into clear diagrams. To couple the whole complete system into subsystems.

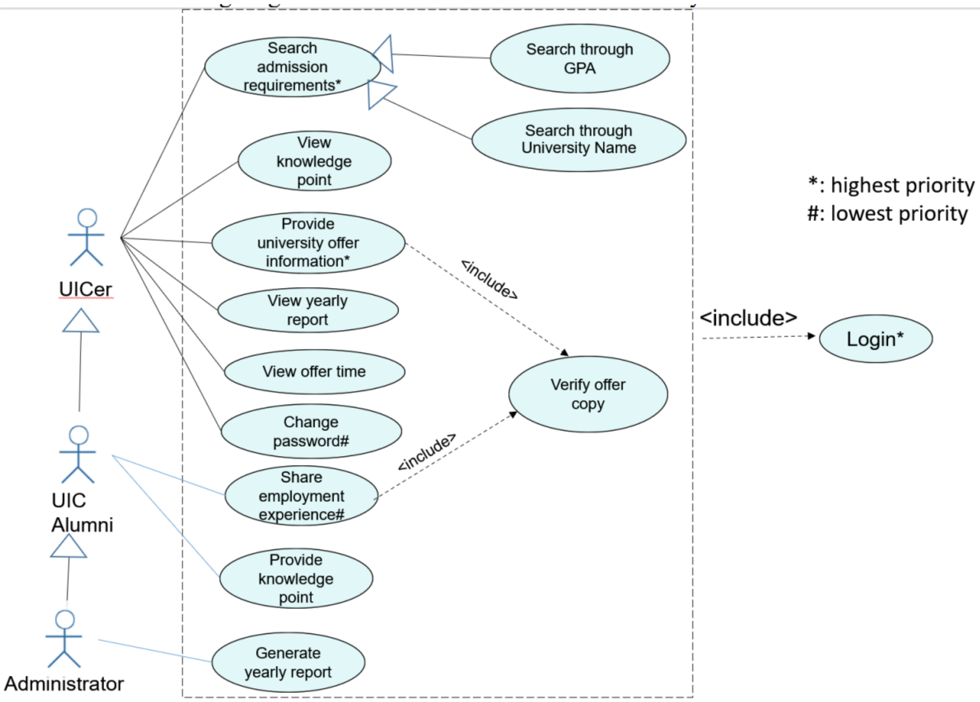
Advantage:

1. Developer can easily understand the components of this software.
2. Avoid confliction and misunderstanding.

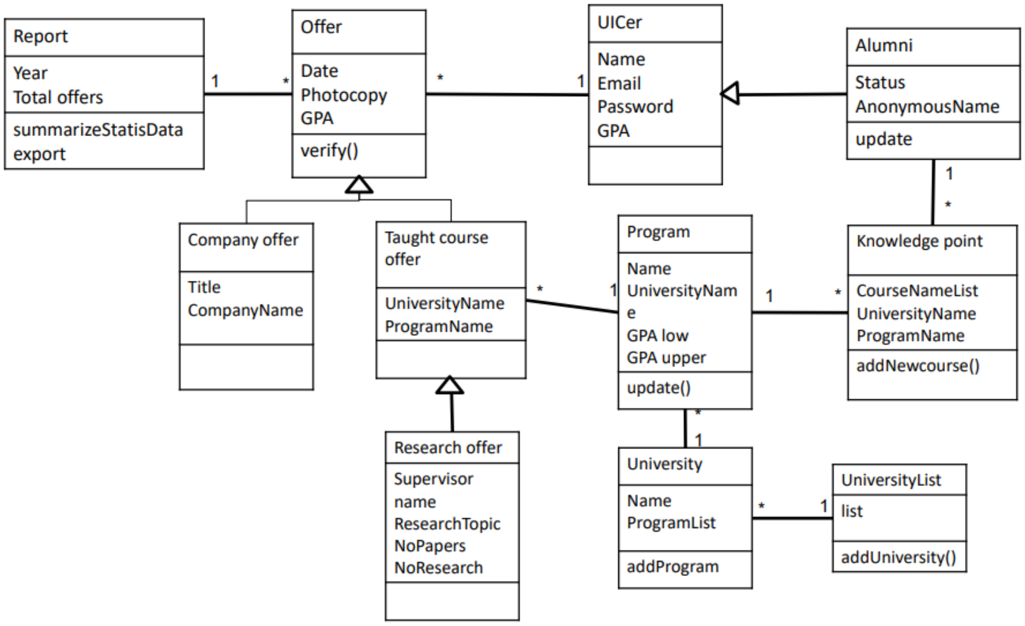
2. Overall description

2.1 Use case diagram and class diagram

Use case diagram



Class Diagram:

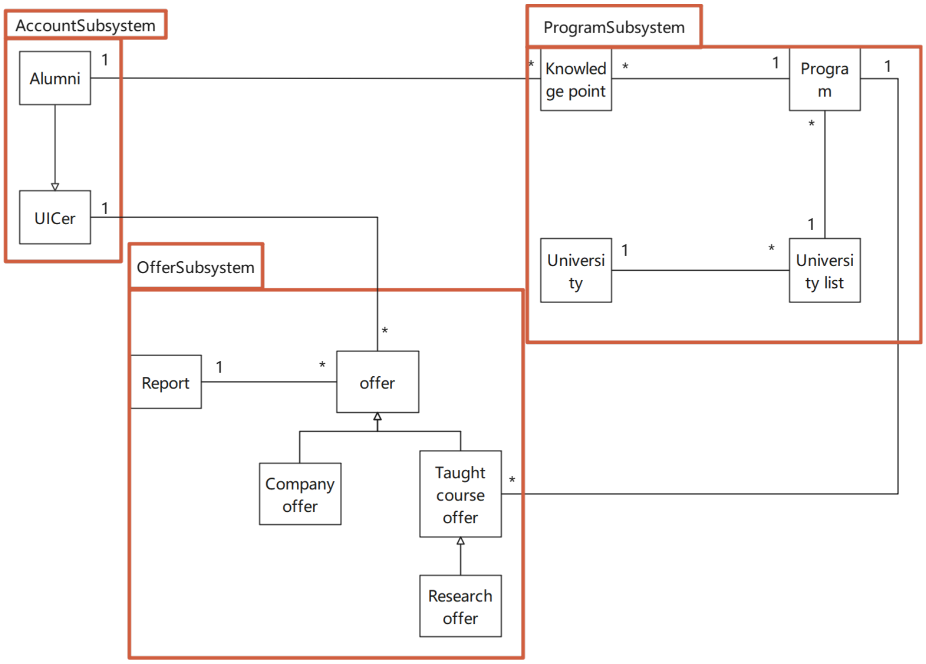


2.2 Design model

We used the layered model - three tier, because it is easier to correlate UI, subsystems and databases.

2.3 System architecture

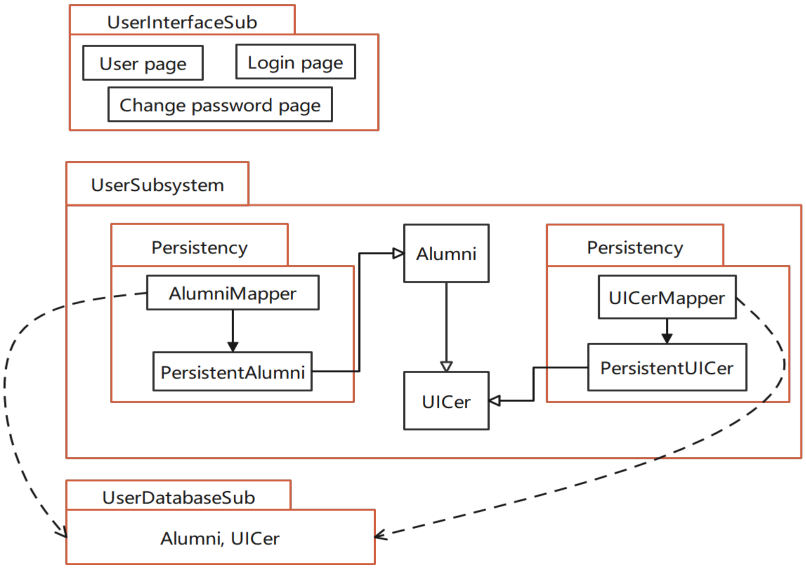
The system is coupled into three subsystems, one is for account, one is for checking and providing offer, and the other one is for finding the university program and college list.



3. System architecture

3.1 User Subsystem

This subsystem includes Account Interface, subsystem design and account databases.
Diagram:



3.1.1 Description

User subsystem Interfaces include: “User page”, “Login Page”, “Change password page”.
User subsystem classes include: “UICer”, “Alumni”.
User database: UICer is persistent class (including its subclass).

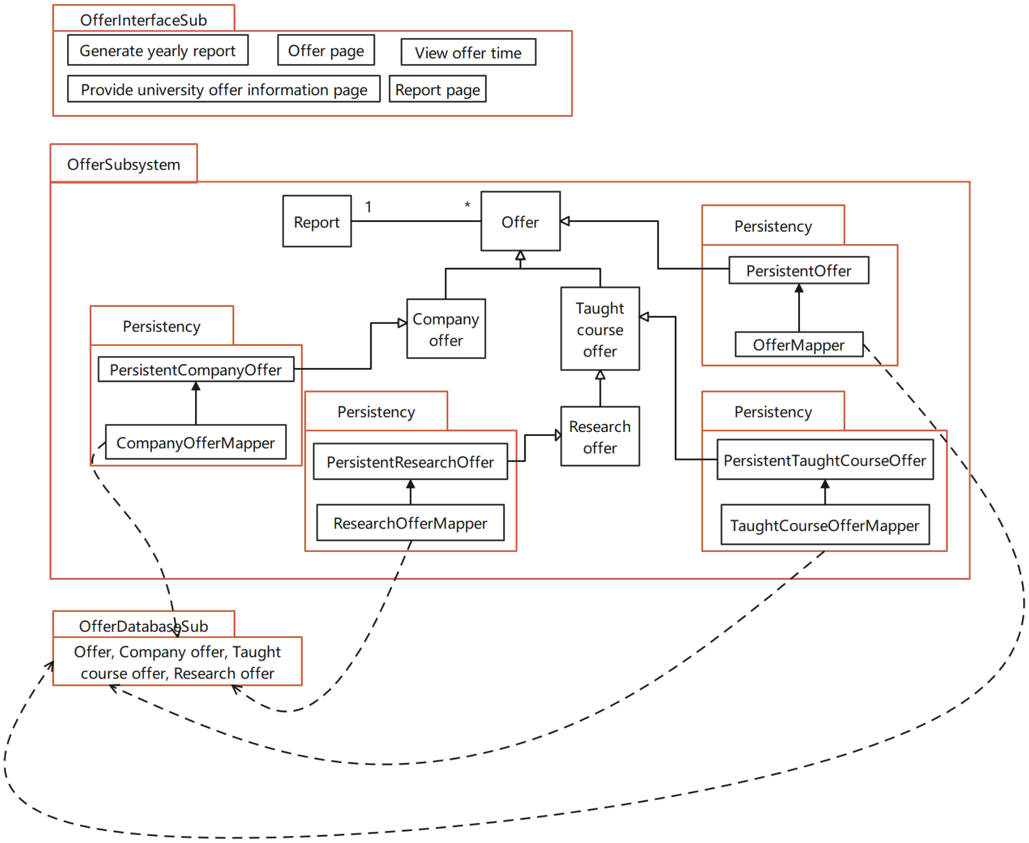
3.1.2 Database

UICer			
Name	Password	GPA	Email

Alumni					
Name	Password	GPA	Email (For association)	Status	AnonymousName

3.2 Offer Subsystem

Offer subsystem includes offer interfaces, offer subsystem design and offer databases.
Diagram:



3.2.1 Description

Offer subsystem Interfaces include: “Generate yearly report”, “Offer page”, “View offer time”, “Provide university offer information page”, “Report Page”, “Provide university offer information”.

Offer subsystem classes include: “Offer”, “Report”, “Company offer”, “Taught course offer”, “Research offer”.

Offer database: “Offer” is persistent class (including its subclasses).

3.2.2 Database

Offer		
Date	Photocopy	GPA

Company offer				
Date	Photocopy (For association)	GPA	Title	CompanyName

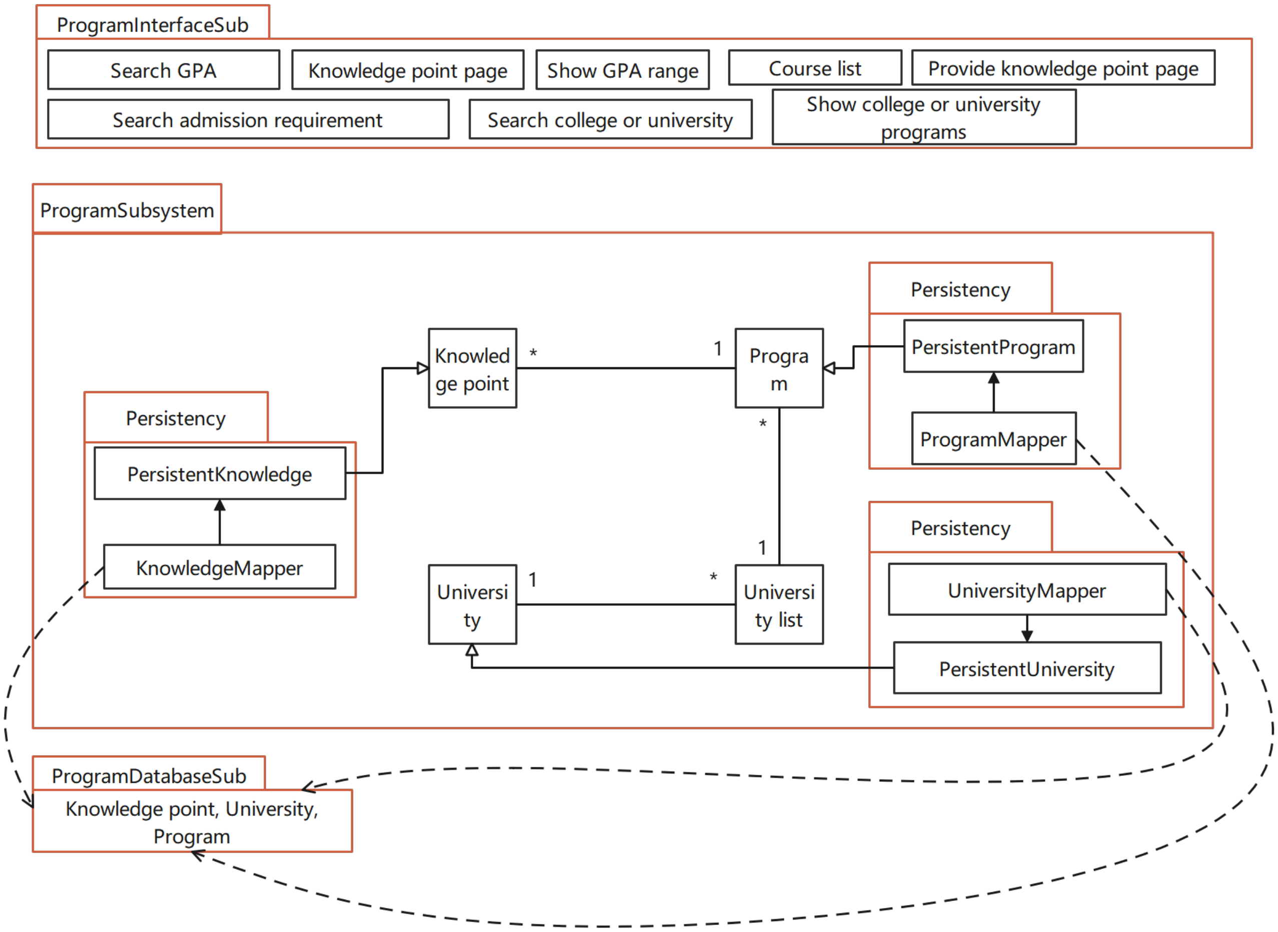
Taught course offer				
Date	Photocopy (For association)	GPA	ProgramName	UniversityName

Research offer									
Date	Photocopy (For association)	GPA	ProgramName	UniversityName	Supervisor	name	ResearchTopic	NoPapers	NoResearch

……

3.3 Program Subsystem

Program subsystem includes program interfaces, program subsystem design and program databases.
Diagram:



3.3.1 Description

Program subsystem Interfaces include: “Search GPA”, “Search admission requirement”, “Show college or university programs”, “Search college or university”, “Show GPA range”, “Course list”, “Knowledge point page”, “Provide knowledge point page”.

Program subsystem classes include: “Knowledge point”, “Program”, “University list”, “University”.

Program database: “Offer” and “Report” is persistent class.

3.3.2 Database

University	
Name (for asssociation)	ProgramList

Program			
Name	UniversityName (for association)	GPA low	GPA upper

Knowledge point		
CourseNameList	UniversityName (for association)	ProgramName

4. Assessment

4.1 Stability

Subsystems run separately. Every subsystem is independent, it will not influence others.

4.2 Reusability

A layer can be replaced by another one if the interface is not changed. If the classes in other place are same as the classes in our architecture, then our architecture can be used without changes.

4.3 Scalability

We use Layered Model (three-tier architecture), it is hard to structure into layers and the performance can be a problem.

5. Alternative design (optional)

6. More considerations

TBD.

7. Appendix