UNIVERSITY OF WOLLONGONG AUSTRALIA

School of Computing and Information Technology

Object Oriented Design and Programming

Subject code: CSIT121

Assignment 2 - Autumn Session 2024

Instructor: Dr Tianbing Xia, Dr Fenghui Ren, and Mr Kevin Malysiak

Student name: Nguyen Van Binh

Student number: 8568418

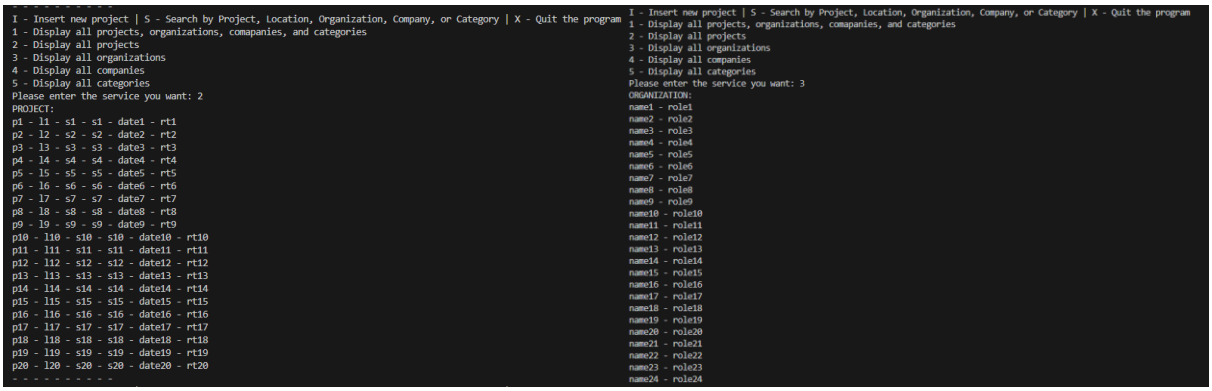
Topic: A program allows users to input new projects, organizations, companies, and categories by an input file and export the result to an output file by I/O file mechanism and can handle Python exceptions.

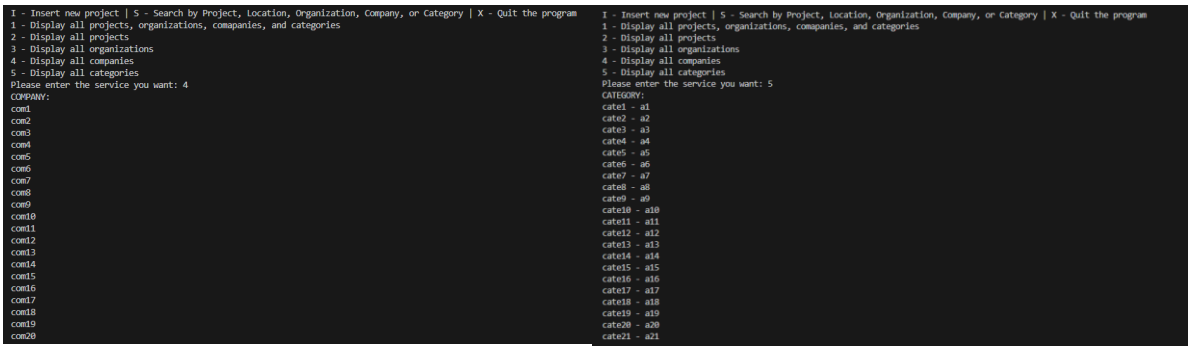
```
PS C:\Users\VanBinh\OneDrive - University of Wollongong\2024_Autumn\Object Oriented Design and Programming\Assignment\
I - Insert new project | S - Search by Project, Location, Organization, Company, or Category | X - Quit the program
1 - Display all projects, organizations, comapanies, and categories
2 - Display all projects
3 - Display all organizations
4 - Display all companies
5 - Display all categories
Please enter the service you want:
```

At the beginning of the program, the program has 8 functions such as “Insert new project”, “Search by project title, location, organization name and role, company, category name, and achievement”, “Quit the program”, “Display all projects, organizations, companies, and categories”, “Display all projects”, “Display all organizations”, “Display all companies”, and “Display all categories”. Users can choose a function by clicking the letter and number that is written before the function (including upper cases and lower cases).

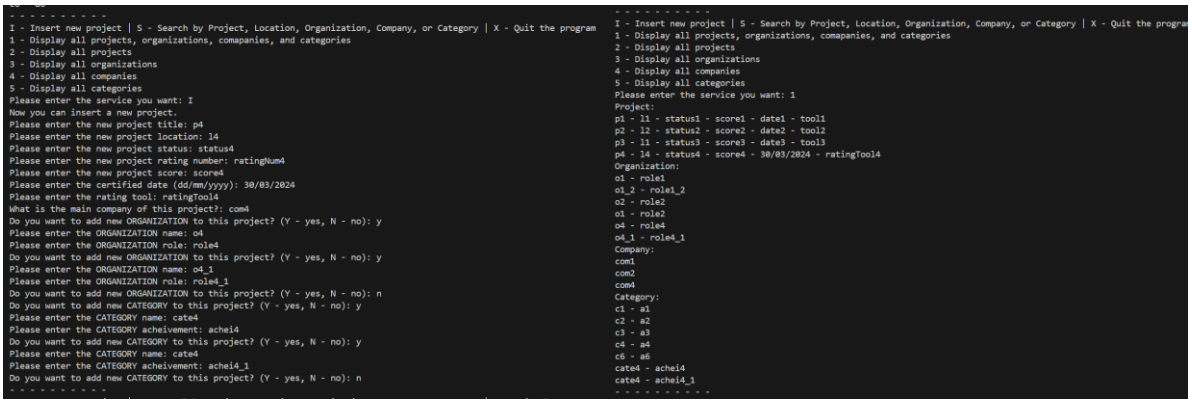
I created the first sample and then used ChatGPT to create 20 samples with different attributes (I modified some samples to make them similar in some attributes to have dynamic and interesting results when using the searching function) in the input file “input.txt”, the user can modify any attributes in the project, and add or remove projects, organizations, companies, or categories,… in the input.txt. The program will read and save the data of input.txt to the system to serve the functions of the system.

```
PS C:\Users\VanBinh\OneDrive - University of Wollongong\2024_Autumn\Object Oriented Design and Programming\Assignment\Ass2> p
I - Insert new project | S - Search by Project, Location, Organization, Company, or Category | X - Quit the program
1 - Display all projects, organizations, comapanies, and categories
2 - Display all projects
3 - Display all organizations
4 - Display all companies
5 - Display all categories
Please enter the service you want: 1
PROJECT:
p1 - l1 - s1 - s1 - date1 - rt1
p2 - l2 - s2 - s2 - date2 - rt2
p3 - l3 - s3 - s3 - date3 - rt3
p4 - l4 - s4 - s4 - date4 - rt4
p5 - l5 - s5 - s5 - date5 - rt5
p6 - l6 - s6 - s6 - date6 - rt6
p7 - l7 - s7 - s7 - date7 - rt7
p8 - l8 - s8 - s8 - date8 - rt8
p9 - l9 - s9 - s9 - date9 - rt9
p10 - l10 - s10 - s10 - date10 - rt10
p11 - l11 - s11 - s11 - date11 - rt11
p12 - l12 - s12 - s12 - date12 - rt12
p13 - l13 - s13 - s13 - date13 - rt13
p14 - l14 - s14 - s14 - date14 - rt14
p15 - l15 - s15 - s15 - date15 - rt15
p16 - l16 - s16 - s16 - date16 - rt16
p17 - l17 - s17 - s17 - date17 - rt17
p18 - l18 - s18 - s18 - date18 - rt18
p19 - l19 - s19 - s19 - date19 - rt19
p20 - l20 - s20 - s20 - date20 - rt20
ORGANIZATION:
name1 - role1
name2 - role2
name3 - role3
name4 - role4
name5 - role5
```

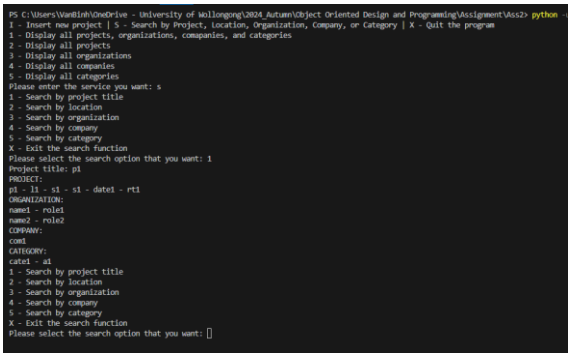


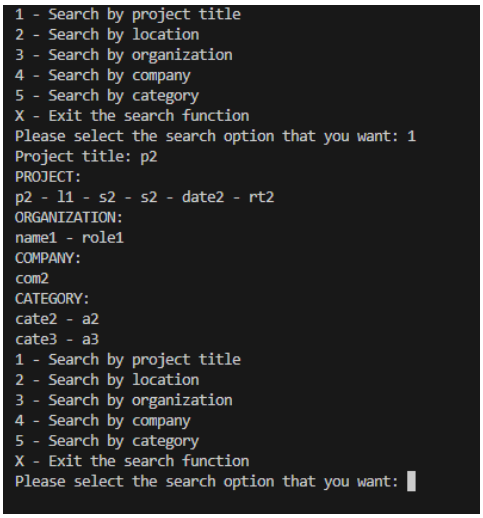


These 5 pictures are 5 functions that are used to display projects, organizations, companies, and categories… depending on the requirement of the user.

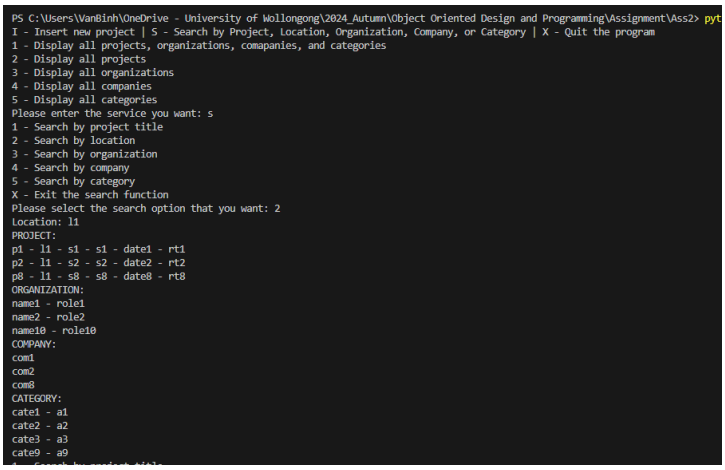


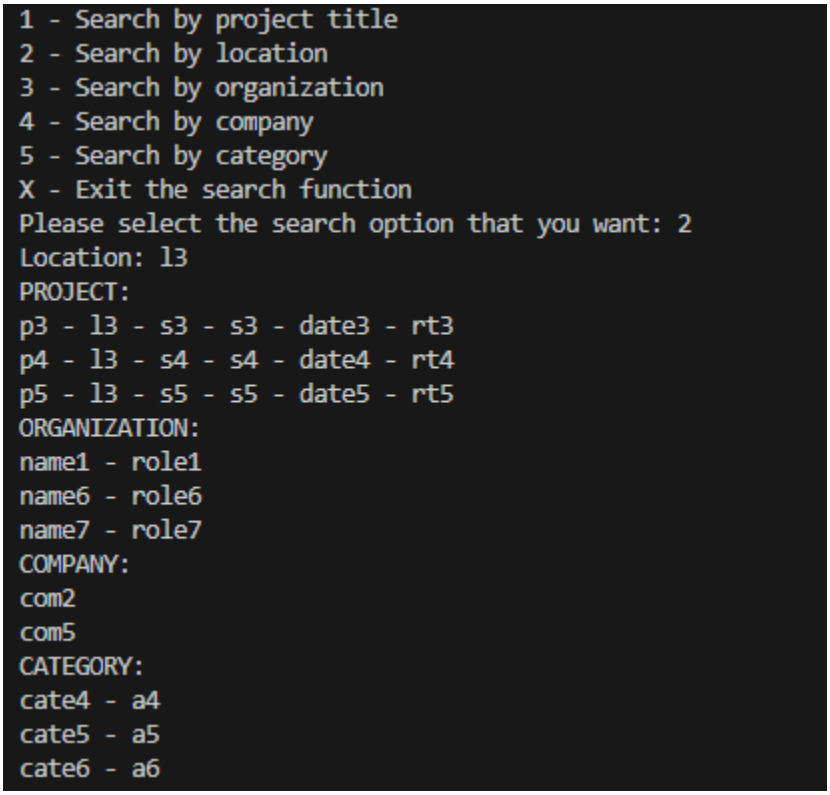
After that, I will use the function “Insert” and input the data in the first picture. When completing the “Insert” function I use the “Display all projects, organizations…” function to check whether my data from input had been created and added to the program.

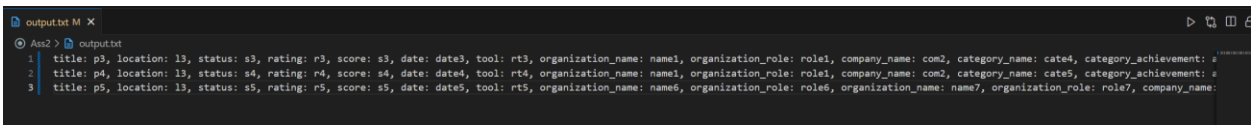




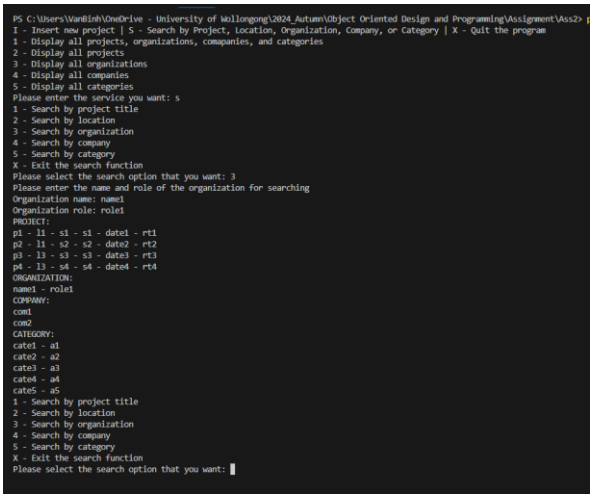
As the picture I use search by project title and search the title “p1” and “p2” respectively. The output will be displayed on the terminal and txt file (output.txt) with 2 different forms).

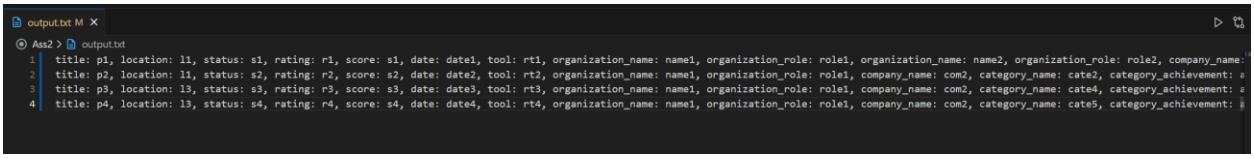




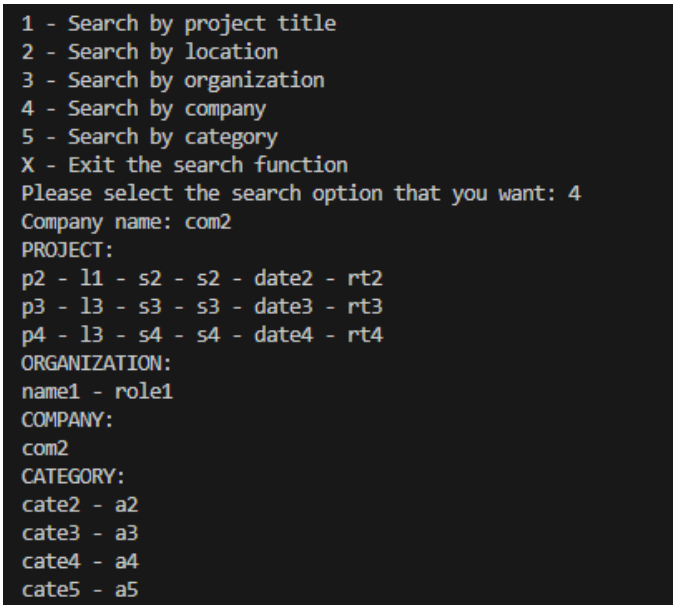


As in the picture, the program searches by location and searches the location “l1” and l3” respectively.



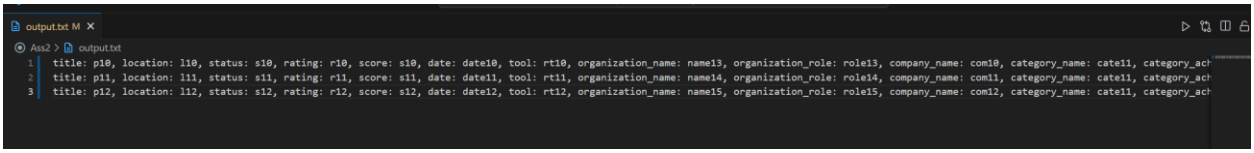


As in the picture, the program searches by organization with its name and role are “name1” and role1” respectively.



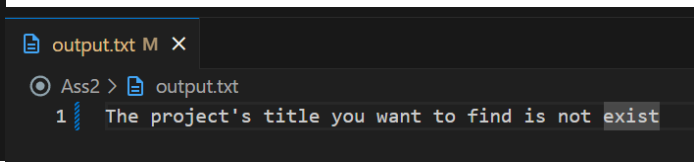
As in the picture, the program searches by company with its name is “com1”.

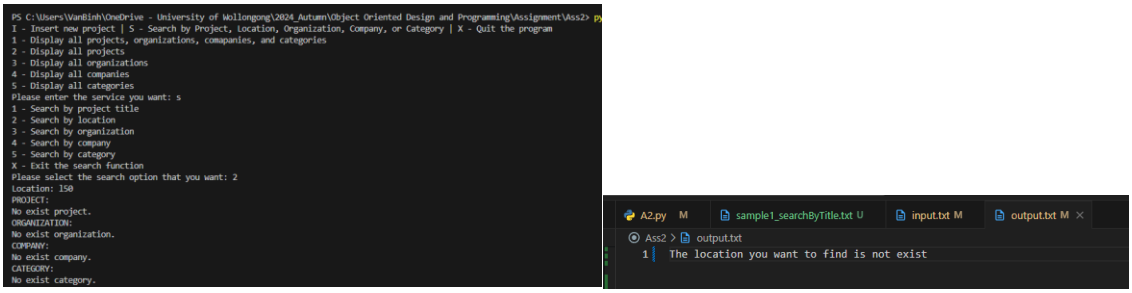
```
1 - Search by project title
2 - Search by location
3 - Search by organization
4 - Search by company
5 - Search by category
X - Exit the search function
Please select the search option that you want: 5
Category name: cate11
Category achievement: a11
PROJECT:
p10 - l10 - s10 - s10 - date10 - rt10
p11 - l11 - s11 - s11 - date11 - rt11
p12 - l12 - s12 - s12 - date12 - rt12
ORGANIZATION:
name13 - role13
name14 - role14
name15 - role15
COMPANY:
com10
com11
com12
CATEGORY:
cate11 - a11
```



As in the picture, the program searches by category with its name and achievement are “cate11” and “a11” respectively.

```
1 - Search by project title
2 - Search by location
3 - Search by organization
4 - Search by company
5 - Search by category
X - Exit the search function
Please select the search option that you want: 1
Project title: p100
PROJECT:
No exist project.
ORGANIZATION:
No exist organization.
COMPANY:
No exist company.
CATEGORY:
No exist category.
```





These are some examples the user inputs project title, location, or company that does not exist and the program cannot find relevant information on the data the user inputs.

```
PS C:\Users\VanBinh\OneDrive - University of Wollongong\2024_Autumn\Object Oriented Design and Programming\Assignment\Ass2> p
- University of Wollongong\2024_Autumn\Object Oriented Design and Programming\Assignment\Ass2\A2.py"
At line 2: ERROR when importing project: Project's title already exists !!!
At line 3: ERROR when importing project: Project's title is not valid !!!
At line 7: ERROR when importing project: Project's title is not valid !!!
At line 11: ERROR when importing project: Project's title already exists !!!
```

A2.py M × input.txt M ×

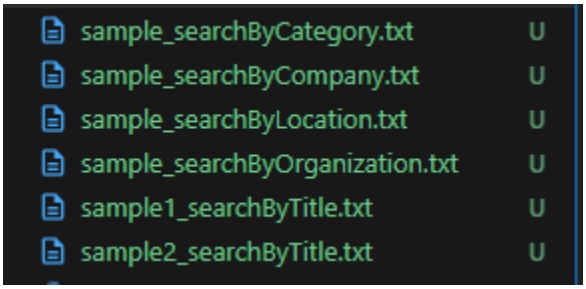
Ass2 > input.txt

```
1  title: p1, location: l1, status: s1, rating: r1, score: s1, date: date1, tool: rt1, organization_name: name1
2  title: p1, location: l1, status: s2, rating: r2, score: s2, date: date2, tool: rt2, organization_name: name1
3  title: , location: l3, status: s3, rating: r3, score: s3, date: date3, tool: rt3, organization_name: name1,
4  title: p4, location: l3, status: s4, rating: r4, score: s4, date: date4, tool: rt4, organization_name: name1
5  title: p5, location: l3, status: s5, rating: r5, score: s5, date: date5, tool: rt5, organization_name: name6
6  title: p6, location: l6, status: s6, rating: r6, score: s6, date: date6, tool: rt6, organization_name: name8
7  title: , location: l7, status: s7, rating: r7, score: s7, date: date7, tool: rt7, organization_name: name9,
8  title: p8, location: l1, status: s8, rating: r8, score: s8, date: date8, tool: rt8, organization_name: name1
9  title: p9, location: l9, status: s9, rating: r9, score: s9, date: date9, tool: rt9, organization_name: name1
10 title: p10, location: l10, status: s10, rating: r10, score: s10, date: date10, tool: rt10, organization_name
11 title: p4, location: l11, status: s11, rating: r11, score: s11, date: date11, tool: rt11, organization_name:
12 title: p12, location: l12, status: s12, rating: r12, score: s12, date: date12, tool: rt12, organization_name
13 title: p13, location: l13, status: s13, rating: r13, score: s13, date: date13, tool: rt13, organization_name
```

The input.txt has some errors like the project has the same title with existing one or some titles are empty so the program still run and have some exceptions of those lines to eliminate them from import data to system.

```
PS C:\Users\VanBinh\OneDrive - University of Wollongong\2024_Autumn\Object Oriented Design and Programming\Assignment\Ass2> pytho
......
----------------------------------------------------------------------
Ran 6 tests in 0.014s

OK
PS C:\Users\VanBinh\OneDrive - University of Wollongong\2024_Autumn\Object Oriented Design and Programming\Assignment\Ass2>
```

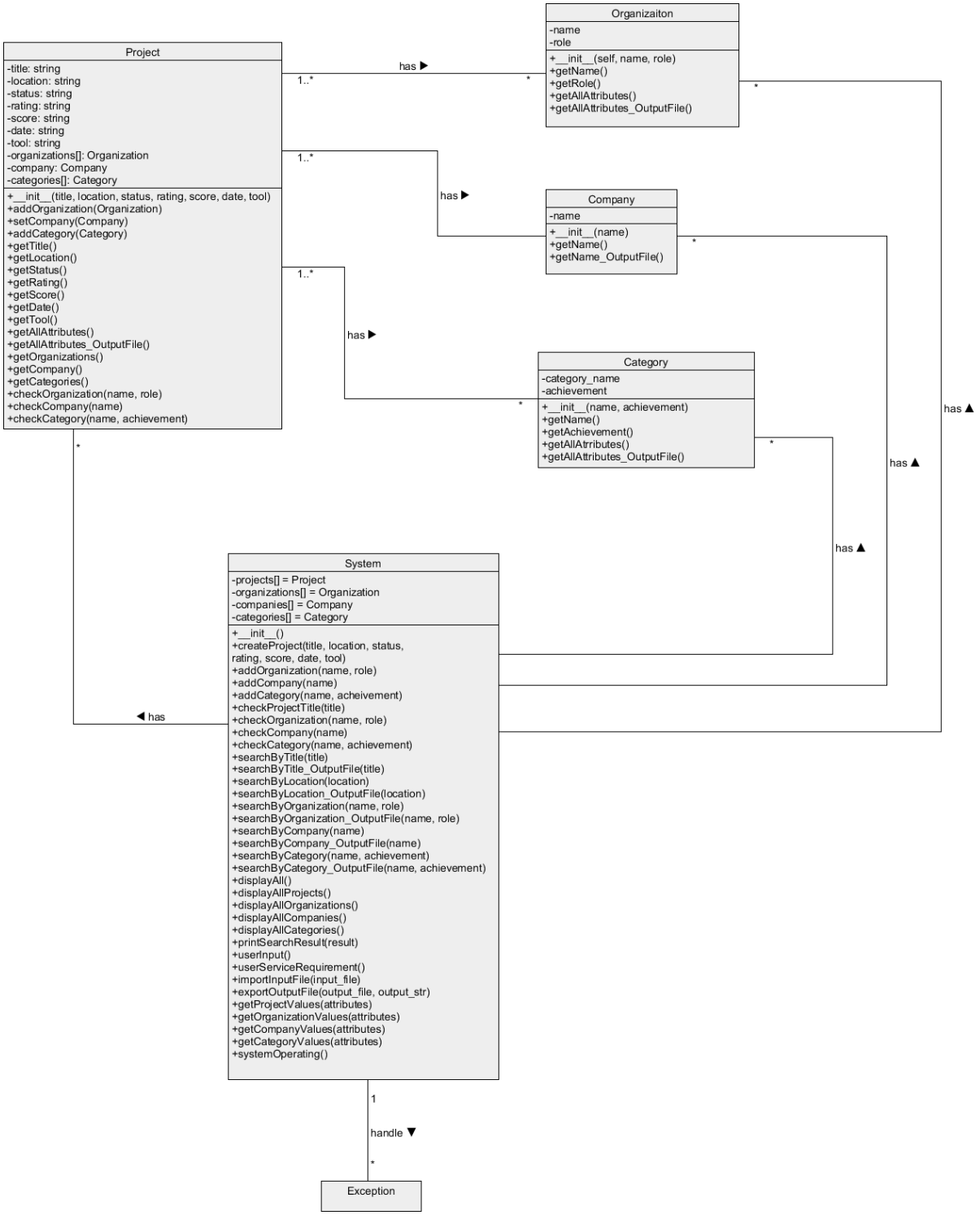


I created some sample in txt file to run unittest. To implement the A2_unittest.py, let comment from lines 779-782. I make test cases for all “Search” functions.

```
PS C:\Users\VanBinh\OneDrive - University of Wollongong\2024_Autumn\Object Oriented Design and Programming\Assignment\Ass2> python -m coverage run A2_unittest.py
......
----------------------------------------------------------------------
Ran 6 tests in 0.040s

OK
PS C:\Users\VanBinh\OneDrive - University of Wollongong\2024_Autumn\Object Oriented Design and Programming\Assignment\Ass2> python -m coverage report
Name             Stmts   Miss  Cover
------------------------------------
A2.py              631    266    58%
A2_unittest.py      65      0   100%
------------------------------------
TOTAL              696    266    62%
PS C:\Users\VanBinh\OneDrive - University of Wollongong\2024_Autumn\Object Oriented Design and Programming\Assignment\Ass2>
```

This is report which is created by coverage command.


Organizaiton
-name
-role
+__init__(self, name, role)
+getName()
+getRole()
+getAllAttributes()
+getAllAttributes_OutputFile()
Project
-title: string
-location: string
-status: string
-rating: string
-score: string
-date: string
-tool: string
-organizations[]: Organization
-company: Company
-categories[]: Category
+__init__(title, location, status, rating, score, date, tool)
+addOrganization(Organization)
+setCompany(Company)
+addCategory(Category)
+getTitle()
+getLocation()
+getStatus()
+getRating()
+getScore()
+getDate()
+getTool()
+getAllAttributes()
+getAllAttributes_OutputFile()
+getOrganizations()
+getCompany()
+getCategories()
+checkOrganization(name, role)
+checkCompany(name)
+checkCategory(name, achievement)
has ▶
1..*
*
1..*
has ▶
Company
-name
+__init__(name)
+getName()
+getName_OutputFile()
*
1..*
has ▶
Category
-category_name
-achievement
+__init__(name, achievement)
+getName()
+getAchievement()
+getAllAttributes()
+getAllAttributes_OutputFile()
*
*
has ▲
has ▲
has ▲
*
◀ has
System
-projects[] = Project
-organizations[] = Organization
-companies[] = Company
-categories[] = Category
+__init__()
+createProject(title, location, status,
rating, score, date, tool)
+addOrganization(name, role)
+addCompany(name)
+addCategory(name, acheivement)
+checkProjectTitle(title)
+checkOrganization(name, role)
+checkCompany(name)
+checkCategory(name, achievement)
+searchByTitle(title)
+searchByTitle_OutputFile(title)
+searchByLocation(location)
+searchByLocation_OutputFile(location)
+searchByOrganization(name, role)
+searchByOrganization_OutputFile(name, role)
+searchByCompany(name)
+searchByCompany_OutputFile(name)
+searchByCategory(name, achievement)
+searchByCategory_OutputFile(name, achievement)
+displayAll()
+displayAllProjects()
+displayAllOrganizations()
+displayAllCompanies()
+displayAllCategories()
+printSearchResult(result)
+userInput()
+userServiceRequirement()
+importInputFile(input_file)
+exportOutputFile(output_file, output_str)
+getProjectValues(attributes)
+getOrganizationValues(attributes)
+getCompanyValues(attributes)
+getCategoryValues(attributes)
+systemOperating()
1
handle ▼
*
Exception

This is the UML diagram for my program. The diagram illustrates the system used to control objects including Projects, Organizations, Company, and Category. All these objects are contained in the array of systems. The system has some functions for creating objects, searching objects by the requirement of the user, and displaying objects. Projects is the general object and has a unique title so it can be identified by its title which contains 3 remaining objects Organization, Company, and Category. If the user set the project's title which same as exist one, the system will requre user to input again. A project can have many organizations and categories but it just can have one company. However, an organization, a company, and a category can belong to many projects.

Project class includes attributes title, location, status, rating, score, date, tool which are string data type, company is Company class, and organizations, categories are a list of Organization class and Category class respectively. Methods addOrganization(Organization), setCompany(Com), and addCategory(Category) used to add objects that belong to the Project. Methods getTitle(), getLocation(), getStatus(), getRating(), getScore(), getDate(), getTool(), getOrganization(), and getCompany(), are used to get the attributes of the Projects class. checkOraganization(name, role), checkCompany(name) and checkCategory(name, achievement) return boolean value, they check whether the object organization or category belongs to the project. Finally, getAllAttributes() is used to concat all attributes of the project and return to print it to the screen, and getAllAttributes_OutputFile() is used to return the concatenation string for displaying in the output-file.

Organization class has attributes including name and role. Method getName() and getRole() return the name and role of the organization object, getAllAttributes() returns the concatenation of all attributes, and getAllAttributes_OutputFile() returns the string to display on the output-file.

Company class just has an attribute name and a method getName().

Organization class has category_name and achievement for attributes. Method getName() and getAchievement() return category_name and achievement, getAllAttributes() returns the concatenation of all attributes of the category, and getAllAttributes_OutputFile() returns the string to display on the output-file.

System class contains attributes that include projects, organizations, companies, and categories which are a list of Project, Organization, Company, and Category. Method createProject(…), addOrganization(…), addCompany(…), and addCategory(…) create objects. checkProjectTitle(…) to guarantee the title is unique. 5 methods find all relevant information of the user's requirement. 5 display methods to print the relevant information of requirement. printSearchResult() displays the list of results that are gained from search functions. userInput() prints the service for the user and gets the information of the user to create the data. The function userServiceRequirement() is used to get the require from the user and for repeat until user enter 'X' or 'x'. systemOperating() contains userServiceRequirement() and have some exceptions to handle errors of the program to help it can run although meet error. importInputFile(…) is used to read and analyze data and put data to the system. getProjectValues(attributes), getOrganizationValues(attributes), getCompanyValues(attributes), and

getCategoryValues(attributes) help the import the input-file. Finally, the exportOutputFile(output_file, output_str) receive the result from the import function and write it again to the output-file.