Due Date: 21.05.2022 23:59

CENG431 – Building Software Systems
Homework 3

In this homework, you are expected to implement an application called "**OpenResearch**" with a GUI. You should fulfill the following concepts:

- SOLID Principles
- MVC Design Patterns
- Java Swing
- Observer Design Pattern

The users of this platform will be scientific researchers. On this platform, the researchers can download the papers to read and create reading lists of papers. Also, the researchers will see other researchers' reading lists.

The GUI of the **OpenResearch** allows researchers to:

- Log in.
- Download a paper by pushing the download button.
- Create a reading list of papers (there can be more than one reading list).
- Add and remove a paper from a reading list.
- Follow/unfollow other researchers.
- View the reading lists of the other researchers.

For this homework, you will be given a folder of papers (Homework3.zip). There will be a PDF file and a BİB file for each paper in this folder. The PDF file stores the content of the paper. The BİB file stores the information of the paper.

In this system, there are two types of papers: Conference Papers and Articles. You can distinguish between the two types of papers in the Homework3.zip folder in the following way:

A. A conference paper has the file name prefix "IP" and its first-line entry is "@inproceedings".
B. An article has the file name prefix "A" and its first-line entry is "@article".

Parse the following information from the BİB files.

1. **For conference papers-> author(s), title, year, DOI, booktitle**
2. **For articles -> author(s), title, year, volume, number, DOI, journal**

After parsing the BİB files in the Homework3.zip folder, create a **papers.csv** file that contains the paper type and the number of downloads for each paper in addition to the information you parsed in bullets 1 and 2. For example, for a conference paper, one of the lines of the CSV file will be "***conference paper, author(s), title, year, doi, booktitle, the number of downloads***". The number of downloads will be determined randomly between 0-1500 before writing on papers.csv.

The created reading lists on this platform should be written on JSON file with the following information:

readinglist_id,creator_researcher_name,readinglist_name,number_of_papers,name_of_papers

The researchers must enter the reading list name as input. Each time a new paper is added to a reading list, the JSON file and the page that shows the reading lists' information should be updated. There should be no duplicate papers in the reading lists.

Every time a researcher pushes the download button to download a paper, the **papers.csv** file should be updated for the number of downloads. Furthermore, if any of these operations affect any of the researchers' reading lists these changes should be applied to these as well.

The researchers' information is stored in an XML file with the following information:

researcher_name,password,following_ researcher_names,follower_ researcher_names

When a researcher performs follow/unfollow operation the XML file should be updated. You should create an XML file that consists of at least 5 researchers including the instructor and the teaching assistants of this course. **The reading lists, following researchers and follower researchers should be displayed on a researcher's profile page.**

The BIB files and the XML file should be loaded first when the program starts and then the CSV file should be created accordingly. After a researcher creates a reading list, the JSON file should be created too, and it should be updated when necessary.

You may use the PDF files in Homework3.zip to show the papers in your system GUI (this is an optional requirement, not a must-have).

You can use external libraries to parse or create any type of file (BIB, XML, CSV, JSON, PDF). (Please, read the submission rules of external libraries below.)

Please, implement your project by following SOLID principles and try to increase reusability.

NOTE: Please try your best to make your software user-friendly and make sure it is expressive enough to the user. Some example GUI screens are given on the last page, you can use them as a guideline, you do not have to design and implement the same GUI.

Usability Testing: If you have enough time, you can create a checklist of every functionality that your software has and ask your friends to perform each functionality without giving them any hint. If your friends succeed in every functionality, then your software's usability is quite high.

Project Submission Rules

1. Cheating is not allowed. If any cheating has been detected, they will be graded with 0 and no further discussion will be discussed.
2. You are expected to submit your project in groups. Therefore, only one of you will be sufficient to submit your project.
3. Make sure you export your project as an Eclipse project. You can use other IDEs as well; however, you must test **if it is supported by Eclipse**.

4. If you are using an external library, make sure that the ".jar" library is in your project after you exported it. Unfortunately, from our previous experiences, we have encountered project submissions that use libraries from their "**Desktop**".
5. Please submit your project through Cloud-LMS (https://cloud-lms.iyte.edu.tr/).
6. Please export your Java Project in the given format with your assigned.
   Example:
   G02_CENG431_Homework03.zip. (Your group IDs will be announced on Microsoft Teams).

7. Please be informed that your submissions may be anonymously used in software testing and maintenance research studies. Your names and student IDs will be replaced with non-identifying strings. If you do not want your submissions to be used in research studies, please inform the instructor (Dr. Tuglular) via e-mail.