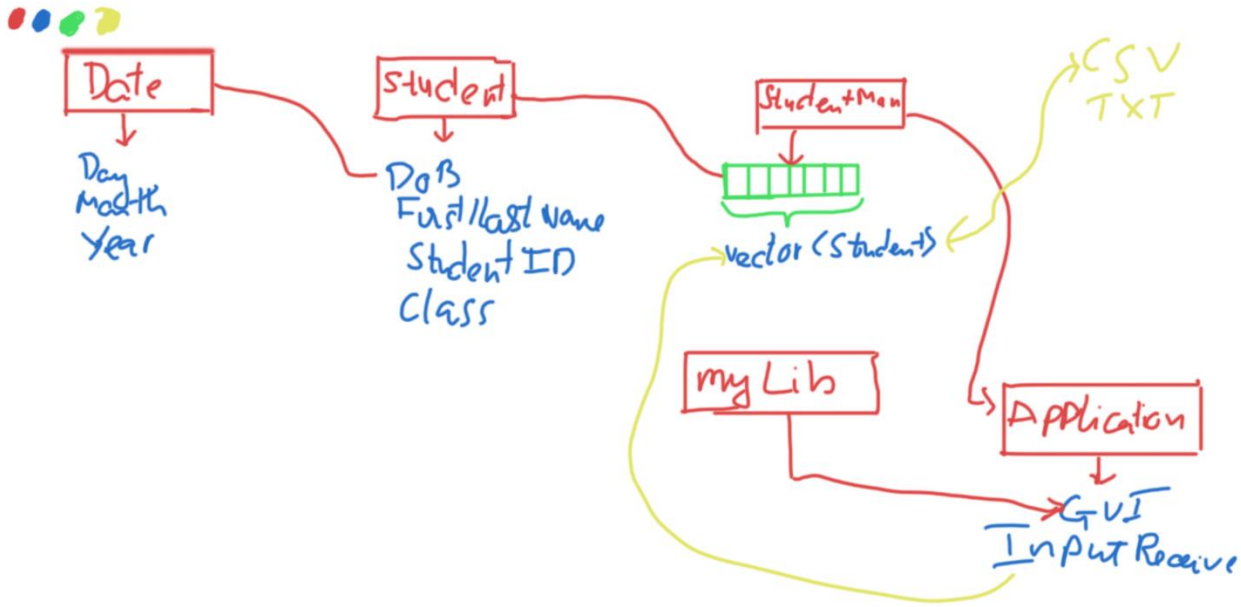
Student Manager

(small scale)

Student: Ta Nguyen Ngoc

Dao Tuan Trung

Teacher: Mai Thuy Nga



Initial draft of the application

So the base idea is this:

- Whenever there is an object needs to be processed more than 3 of its variables, we turn it to a class, so here we will have 3 classes:
    - **Date:** (Vars: day, month, year) which all 3 are related to each other (leap year, 30/31 days, etc,…). Date has these main function:
        - **Create** Day Instance with provided parameters (Constructors).
        - **Process** parameter be for assigning to Date's properties.
    - **Student:** (Vars: Student ID, First/Last Name, Date of Birth (***Date***), Class, which have these main functions (some other supplemental function we will not talk about):
        - **Create** Student Instance with provided parameters.
        - **Create** Student Instance by demanding input from user.
        - **Process** parameters before assigning to Student's properties. (ID/ Class/ DoB (call Date func) should have correct patterns, Name should not have weird characters).
    - **Student Manager:** (A dynamic array (vector) of Student)
        - **Why Vector not other Data Structure?**
            - **Vector vs Array**: Vector support dynamic array. Instead of using array and allocate in heap memory, vector has its own algorithm to optimize these processes.
            - **Vector vs Linked List**: Vector has access member of O(1), instead of O(n) with Linked list. And we only insert member at the end of vector takes linear complexity also so it should not be too slow.
            - **Vector vs Trees**: These students should be linearly (kind of) sorted. Implementing these other DS is overkill and not necessary.
        - **Main functions**:

        - **Import Student** to vector
        - **Sort** list
        - **Delete Student**
        - **Find Student**
        - **Import/Export** from/to TXT/CSV
- There’s also the Application’s main function which serves the following:
    - **Having** Student Manager Instance.
    - **Calling** function of the Student Manger, which leads to Student’s function, Date’s function.
    - **Receive** User Input.
    - **Calling** to our own implemented myLib which serves for GUI.
    - **Automatically** import pre-defined TXT, CSV files.