Iteration 2

Deadline: February 6th

CSE3063F20P2_RAD_GRP12 _iteration2.pdf

Contents

1. Introduction

1.1 Purpose

After the current world events that occurred in this past year, students and teachers had to switch to online remote education. And for this reason, examinations and tests also became online.
Zoom took the world by storm. Its polls system made it so much easier for teachers to administer quizzes in real time for their students. But ZOOM will only go as far as giving us the polls' reports, it can not grade the students in each class that this system is being used in. This is where our software comes in.

Our project will be a simple software solution that can be used to process the poll reports, and the student list of the class, then it will give a detailed report about how well each student that participated in that poll performed.

Furthermore, this software will be able to track the performance of the students throughout several ZOOM polls that has been carried out during the semester

Our audience will be any educational staff in Marmara that wants to use the ZOOM Polls system to make simple polls or examinations, and would like to observe a detailed report on each of their students.

1.2 Scope

This software product can be used by any educational staff in Marmara who uses the ZOOM polls system.
The software has a functionality of reporting/tracking students' performance in the performed poll.

The goal of our project:

- Provide an easy method to easily review each student's performance in the polls.
- Efficiently maintain a database of students.
- Detect possible cheating attempts.

1.3 References

1. IEEE Computer Society. Software Engineering Standards Committee, & IEEE-SA Standards Board. (1998). *IEEE Recommended Practice for Software Requirements Specifications* (Vol. 830, No. 1998). IEEE.
2. https://classroom.google.com/u/0/c/MTgxNjU0NDIyNDBa
3. https://classroom.google.com/u/0/c/MTgxNjU0NDIyNDBa/m/Mzg4NzE2MTc2OTNa/details
4. https://www.wikipedia.org

1.4 Overview

An overview of everything in the document could be summarized as the following:

- We organized the structure of this document to give the reader an understanding of the problem, then how we are going to fix, then go into detail of the product itself.
- What comes after this section is all concerning the product. We give an overall description of factors that might affect the software solution we are working on, then we talk about its functions, the constraints that would apply on those functions, and finally the assumptions we made based on the user characteristics we have.
- We also talk about the specific requirements including, but not limited to, Functional requirements, Performance Requirements, Design Constraints, Software System Attributes, and some other Requirements.

2. Overall description

2.1 Product perspective

The software is intended to be used alongside ZOOM in order to provide improvements to accessing and analyzing ZOOM polls/quizzes. Therefore, this software is not a standalone piece, and it will have to comply to certain constraints which include:

- User interfaces: the software requires a simple and effective user interface in order for the user to benefit from it as much as possible. This interface should be a lot easier than dealing with CSV files directly.
- Memory: As mentioned, the software will of course consume memory. However, a constraint on this amount of memory needs to be set as this piece of software will be a component of a bigger software. Because if not put, it might take up too much and make the system crash.

The constraints vary, however, they will become more clear with each iteration and with each time we increase the requirements and make them more concrete.

2.2 Product functions

The software first starts by taking in multiple inputs that include:

- A ZOOM poll report file in .CSV format.
- A student list from BYS Marmara University system.
- A single answer key file or a folder containing multiple answer keys.

Then the software identifies the poll as either an attendance poll or a quiz. If the poll is an attendance poll it will add the attendance for each student to an Excel spreadsheet. The mentioned spreadsheet contains the number of attendances for each student for a specific class, and it also calculates the attendance rate for each student.

However, if the poll is of the quiz type it will output two different spreadsheets for each poll. The first one will include all the students and their answers for each question. The second spreadsheet will show statistics about the overall performance of that quiz, including pie charts, histograms, etc.

For each question, the user can output an image that contains a pie chart analyzing the answers given by the students.

For each quiz, the user can view a list of a clustered group of students, which are suspected to be cheating because of their similar answers.

Finally, the program will also output a global spreadsheet that accumulates all the previous quizzes for each student, and it will also show the attendance rate for all students.

2.3 User characteristics

This software can be used by any educational staff in Marmara who uses the ZOOM polls. Our system will make grading and processing polls easier, which is important for a lot of education staff nowadays with education being online. The users do not need any pre-training to be able to use our program. After installing required libraries they only need to input a dataset in the expected format and start using our software right away.

2.4 Constraints

- Our program only accepts input datasets of the format .CSV and .XLS.
- The ZOOM poll reports and the student info dataset should match the specific format and the structure that program requires (course name, professor, student list, and so on.

2.5 Assumptions and dependencies

- How big the student lists are.
- Requirements of our software might change, since the teaching staff might need to use multiple spreadsheets for each student instead of one.

3. Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

Our current UI that is shown to the customer, will allow him to import/export the data to/from the program by clicking on the specific buttons

3.1.2 Hardware interfaces

For this iteration the software only requires an operating system with a framework able to run Python programs. For starters, we will limit the devices that will use the software to Computers/Laptops only.

3.1.3 Software interfaces

For this iteration the following software products are required:

1) JSON:
   - Name:
   - Version number: 2.0.9
   - Source: https://docs.python.org/3/library/json.html
   - Purpose: This package is used to manage the database.

2) Pandas:
   - Name: Pandas
   - Version number: 1.2.0
   - Source: https://pandas.pydata.org/
   - Purpose: Easy to use open source data analysis and manipulation tool

3) Matplotlib:
   - Name: Matplotlib
   - Version number: 3.3.3
   - Source: https://matplotlib.org/3.3.3/index.html
   - Purpose: This package is used for visualization in python.
   -

4) PyQT5:
   - Name: PyQT5
   - Version number: 5.12.2

- Source: https://www.riverbankcomputing.com/static/Docs/PyQt5/
- Purpose: This package is used as a python cross-platform GUI toolkit.

5) Xlrd:
    - Name: xlrd
    - Version number: 2.0.1
    - Source: https://xlrd.readthedocs.io/en/latest/
    - Purpose: This package is for reading data and formatting information from older Excel files (ie: .xls).
6) Unittest:
    - Name: unittest
    - Version number: 3.9.1
    - Source: https://github.com/python/cpython/tree/master/Lib/unittest/
    - Purpose: Used to write automated unit tests.
7) Logging:
    - Name: logging
    - Version number: 3.9.1
    - Source: https://github.com/python/cpython/tree/master/Lib/logging
    - Purpose: This module is used for logging events in python.
8) Jdcal:
    - Name: jdcal
    - Version number: https://pypi.org/project/jdcal/
    - Source: 1.4.1
    - Purpose: Used for conversion between julian dates and calendar dates.
9) Pillow:
    - Name: pillow
    - Version number: 8.1.0
    - Source: https://github.com/python-pillow/Pillow/tree/fcc42e0d344146ee9d265d1f43c094ce5a0ec4cf
    - Purpose: This library is used for image processing in python.
10) Et-xmlfile:
    - Name: et-xmlfile
    - Version number: 1.0.1

- Source: https://pypi.org/project/et_xmlfile/
- Purpose: Used for creating large XML files with low memory cost.

11) Kiwisolver:
    - Name: kiwisolver
    - Version number: 1.3.1
    - Source: https://pypi.org/project/kiwisolver/
    - Purpose: Efficient Cassowary constraint solving algorithm

12) Openpyxl:
    - Name: openpyxl
    - Version number: 3.0.5
    - Source: https://openpyxl.readthedocs.io/en/stable/changes.html#id11
    - Purpose: Used to read/write excel files

13) Pyparsing:
    - Name: pyparsing
    - Version number: 2.4.7
    - Source: https://pypi.org/project/pyparsing/
    - Purpose: Text parsing and manipulation

14) Python-dateutil:
    - Name: python-dateutil
    - Version number: 2.8.1
    - Source: https://pypi.org/project/python-dateutil/
    - Purpose: Extension to datetime module with improvements

15) Pytz:
    - Name: pytz
    - Version number: 2020.5
    - Source: https://pypi.org/project/pytz/
    - Purpose: Used to provide a database system.

16) Six:
    - Name: six
    - Version number: 1.15.0
    - Source: https://pypi.org/project/six/
    - Purpose: Provides compatibility between python2 and python3

17) Unidecode:
    - Name: Unidecode
    - Version number: 1.1.2
    - Source: https://pypi.org/project/Unidecode/
    - Purpose: Used to transform text in unicode format to other formats.

3.1.4 Communications interfaces

Within this iteration, the software will run offline so there is no need for any communication interface.

3.2 Functional requirements

3.2.1 Poll Analysis System

Poll Analysis System is the central class in our object-oriented modelling. It is responsible for creating instances of other classes inside itself and call their corresponding operations when necessary.

3.2.1.1 Functional requirement

This class will be responsible for loading all the parsed data, and the exporting of the finished reports.

3.2.2 User Interface

User Interface is a class that contains the overall interface that the user will communicate through with the software.

3.2.2.1 Functional requirement

This class will be responsible for showing the user interface through a python PyQT5 Graphical User Interface, where the user can choose whether to start uploading the files that contain the students list, the answers' key, and the poll reports. Or to export the finalized reports.

3.2.2.2 Functional requirement

From this interface, the user is able to select a specific quiz report and check the analysis for each individual question's answers. Also, the user can export a cheat detection analysis from this interface.

3.2.3 Student List Parser

This class is used for the parsing of our students list file. This class is a “Knowing responsibility” class

3.2.3.1 Functional requirement

This class is responsible for parsing the students information that is inside the uploaded students_list file.
It is also capable of getting a single student from the student list.

3.2.4 Poll Parser

This class is used for the parsing of our poll report files. Along with updating those files.

3.2.4.1 Functional requirement

This class is capable of reading each poll report, and updating it with new information by making a new poll submission for it.

3.2.4.2 Functional requirement

The class can also separate the questions and answers in the poll reports into two different categories instead of their current format. Furthermore, it can later on match a set of questions to their corresponding Answer’s keys.

3.2.5 Answer Key Parser

This class is used for the parsing of our question’s answer key files.

3.2.5.1 Functional requirement

This class is capable of distinguishing between the correct and wrong answers, and it can also get the questions and get the answers separately.

3.2.6 Student

This class is used as a creation tool for the student instances. It’s main purpose is to store the information about the student.

3.2.6.1 Functional requirement

This class stores information about the student like id,name,department, registrations,submissions...etc.

3.2.6.2 Functional requirement

It is also capable of adding a course registration to this current student.

3.2.7 Instructor

This class is used as a creation tool for the instructor instances. It's main purpose is to store the information about the instructor.

3.2.7.1 Functional requirement

This class stores information about the instructor like name,department, and registrations.

3.2.7.2 Functional requirement

It is also capable of adding a course registration to this current instructor.

3.2.8 Department

This class is used as a creation tool for the Department instances. It's main purpose is to store the information about the department.

3.2.8.1 Functional requirement

This class stores information like the instructors that teach in that department, and the students who belong to this department.

3.2.8.2 Functional requirement

This class is also capable of manually adding instructors and students to the Department.

3.2.9 Course

This class is used as a creation tool for the Course instances. Its main purpose is to store the information about the course.

3.2.9.1 Functional requirement

This class stores information like the code, name, credits and registrations it belongs to.

3.2.9.2 Functional requirement

It is also capable of adding registrations to this course.

3.2.10 Registration

This class is used for associating classes with each other. It associates the academic time to the given course, along with which instructor teaches it and which students are enrolled in it.

3.2.10.1 Functional requirement

This class stores information about a course registration that includes the instructor, student, academic semester/year, and the course.

3.2.11 Poll

Poll is a class used as a creation tool for instances of the Polls provided in the parsed polls given by the user. It provides a name, date, time, day, a list of the submissions for this poll, and an answer key.

3.2.11.1 Functional requirement

This class's responsibility is to store the Poll information. It is also capable of adding a new question set with their corresponding answers, and adding a submission to this poll.

3.2.12 Quiz Poll

This class is the same as the Poll class, but it is more specific, it stores the information of a Quiz-type of Polls.

3.2.12.1 Functional requirement

This class's responsibility is to store the Poll information, if it belongs to the Quiz group.

3.2.13 Attendance Poll

This class is the same as the Poll class, but it is more specific, it stores the information of an Attendance-type of Polls.

3.2.13.1 Functional requirement

This class's responsibility is to store the Poll information, if it belongs to the Attendance group.

3.2.14 Poll Submission

Poll Submission is a class that links other classes together, it gives you information about what student took which poll, and what are the questions and answers given to it.

3.2.14.1 Functional requirement

This class's responsibility is to store information about student/poll relations.

3.2.15 Question

Question is a class that is used as a creation tool for instances of the questions provided by the user.

3.2.15.1 Functional requirement

This class's responsibility is to store question information like the text it contains, and the type (multiple choice or not). It can also add the question to a poll submission's questions.

3.2.16 Answer

Answer is a class that is used as a creation tool for instances of the answers provided by the user.

3.2.16.1 Functional requirement

This class's responsibility is to store answer information like the text it contains, and whether it is correct or not. It can also add the answer to a poll submission's answers.

3.2.17 Attendance Report Serializer

This class is used to make a report about the students' attendance.

3.2.17.1 Functional requirement

This class's responsibility is to check how many times a student attended the course, by using the poll submissions by that student on the specified dates of attendance.

3.2.18 Stats Report Serializer

This class is used to make a general report about each quiz's poll information in detail.

3.2.18.1 Functional requirement

This class's responsibility is exporting a report for each quiz, visualizing each question from each quiz and analyzing it, and it is responsible for exporting the global report.

3.2.19 Quiz Cluster Analyzer

This class is used to detect any possible attempt at cheating in the quiz.

3.2.19.1 Functional requirement

This class's responsibility is to visualize a cluster of the same answers given by each student, and group them together as a warning sign for possible cheating amongst the students in a given cluster.

3.5 Other requirements

3.5.1 Non-functional Requirements

3.5.1.1 Usability

The product is very easy to use. The user will get the detailed report about the students' performance only by uploading the required datasets into our system.

3.5.1.2 Reliability

The user can access students' performance report 100% of the time without a failure.

3.5.1.3 Performance

The performance of a product depends on how big the data is. The bigger the dataset, the more time the model takes. Overall the model calculation and response time should be as little as possible.

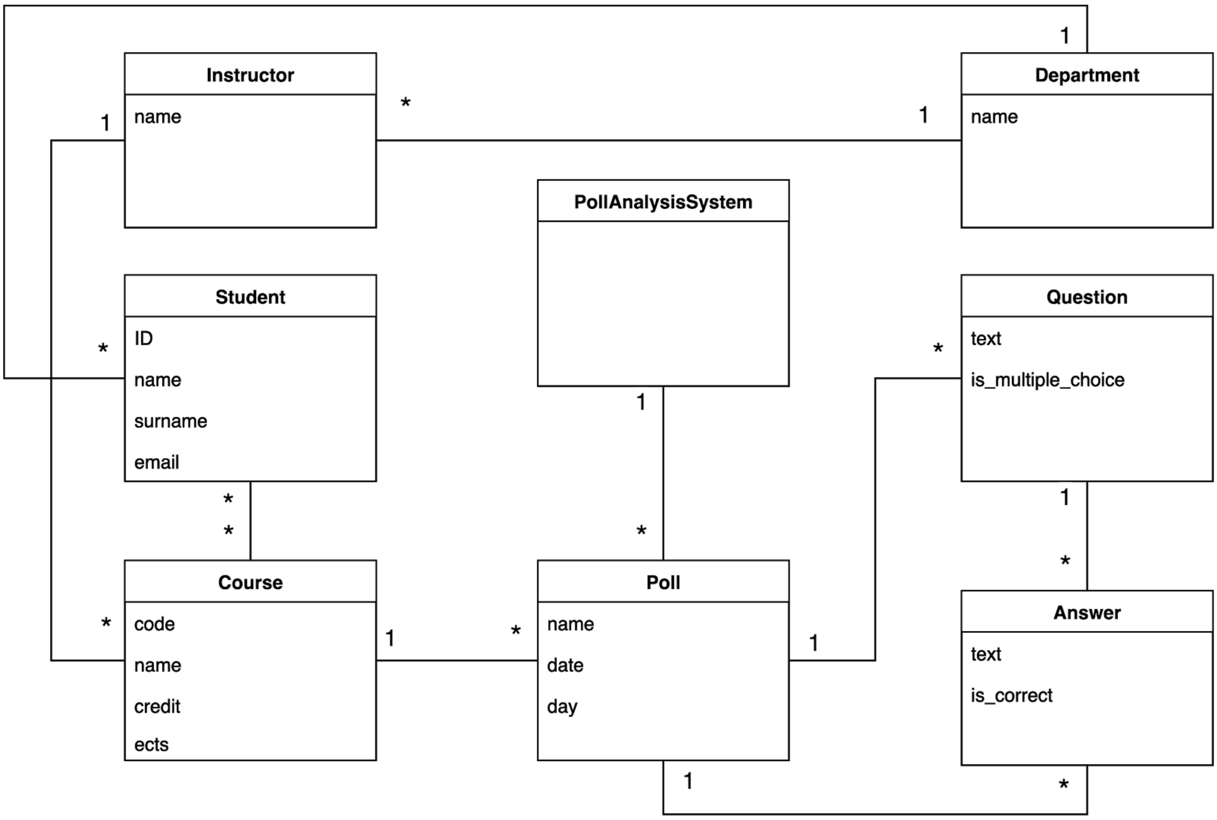
3.5.1.4 Portability

Our product works on all platforms that have terminals and runs python.

3.5.1.5 Availability

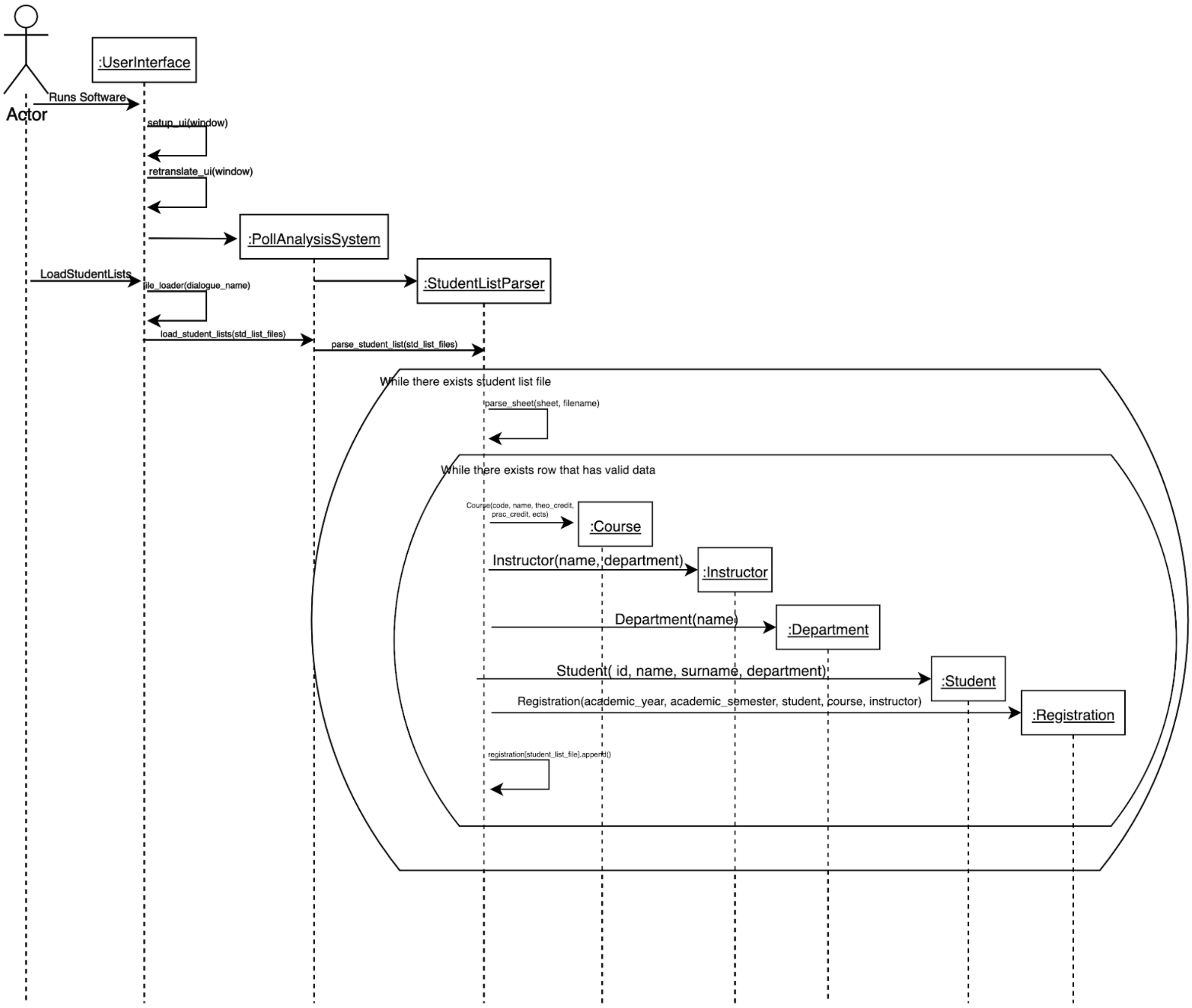
The user can use a software throughout the week at any time during the day.

3.6 Domain Model

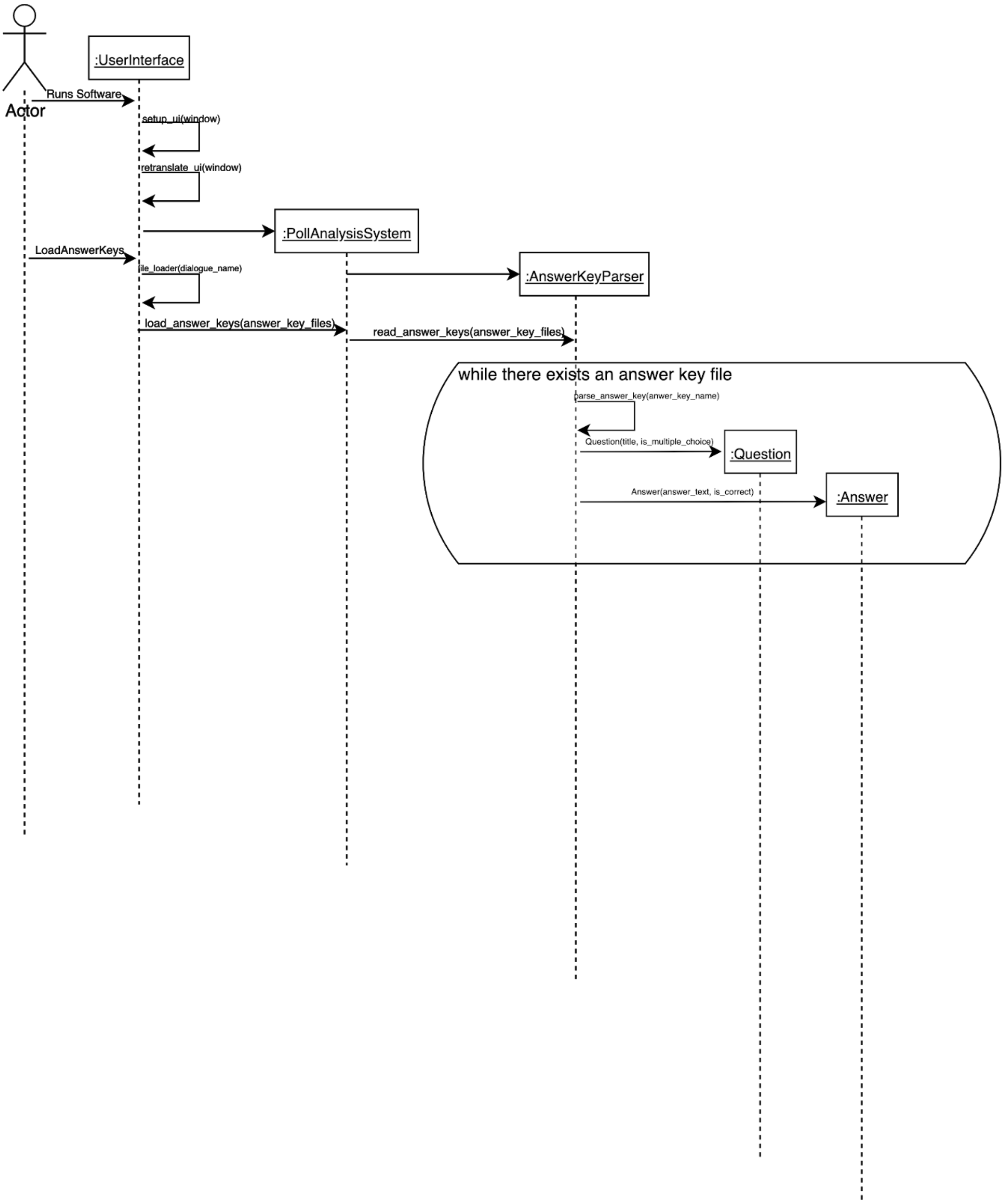


3.7 System Sequence Diagram

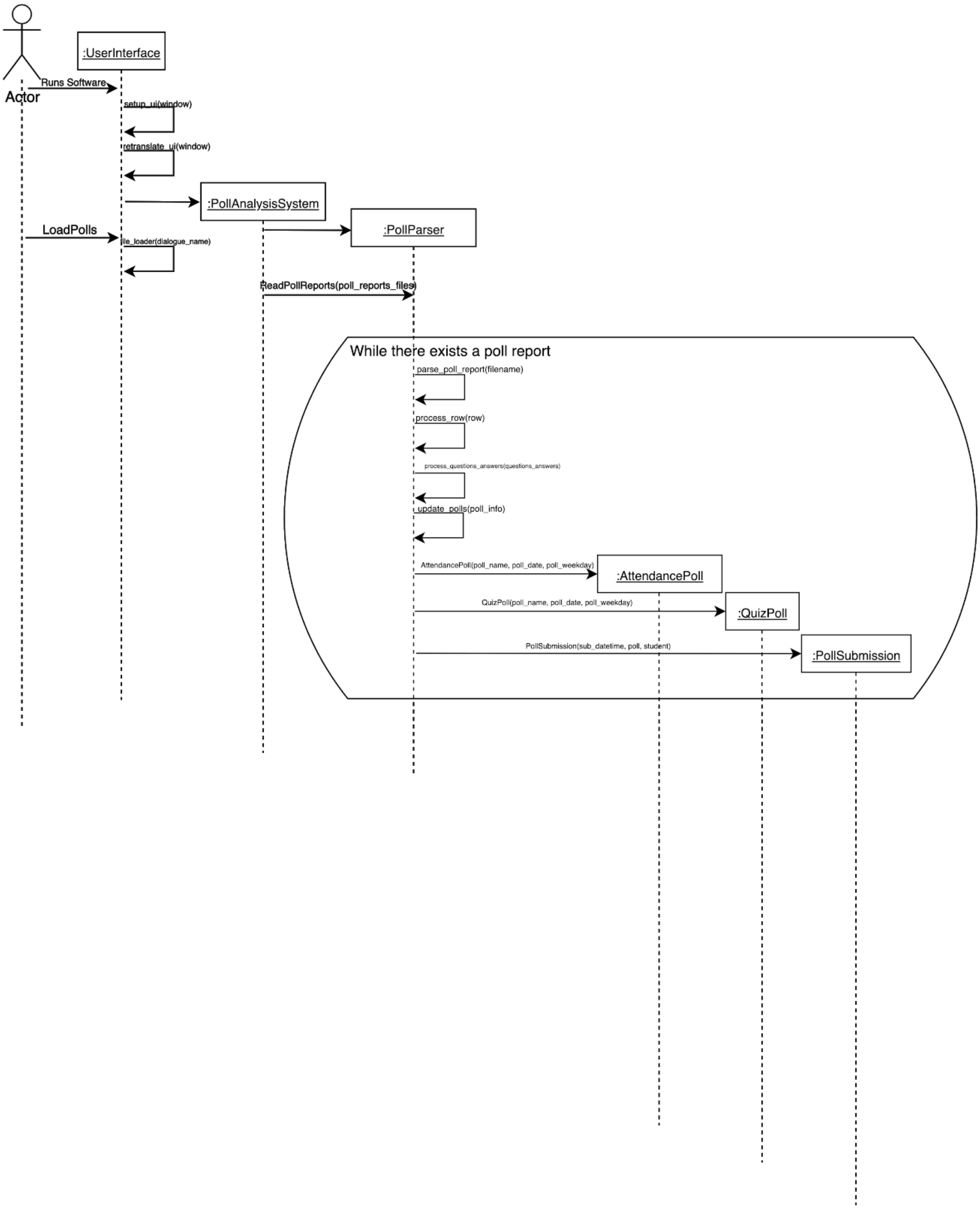
3.7.1 Load Student Lists


Actor
:UserInterface
Runs Software
setup_ui(window)
retranslate_ui(window)
:PollAnalysisSystem
LoadStudentLists
file_loader(dialogue_name)
:StudentListParser
load_student_lists(std_list_files)
parse_student_list(std_list_files)
While there exists student list file
parse_sheet(sheet, filename)
While there exists row that has valid data
Course(code, name, theo_credit, prac_credit, ects)
:Course
Instructor(name, department)
:Instructor
Department(name)
:Department
Student(id, name, surname, department)
:Student
Registration(academic_year, academic_semester, student, course, instructor)
:Registration
registration[student_list_file].append()

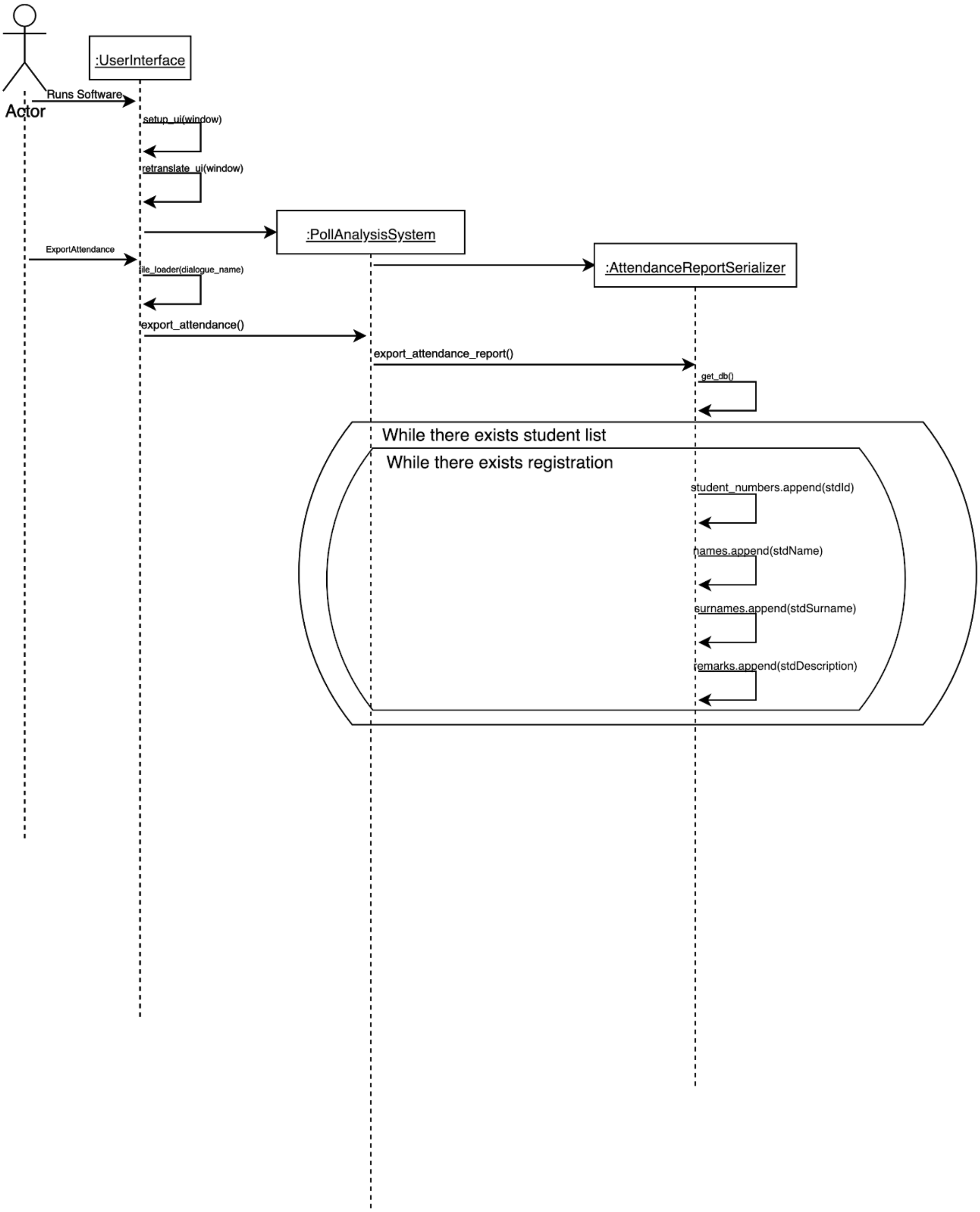
3.7.2 Load Answer Keys


Actor
:UserInterface
Runs Software
setup_ui(window)
retranslate_ui(window)
:PollAnalysisSystem
LoadAnswerKeys
file_loader(dialogue_name)
:AnswerKeyParser
load_answer_keys(answer_key_files)
read_answer_keys(answer_key_files)
while there exists an answer key file
parse_answer_key(anwer_key_name)
Question(title, is_multiple_choice)
:Question
Answer(answer_text, is_correct)
:Answer

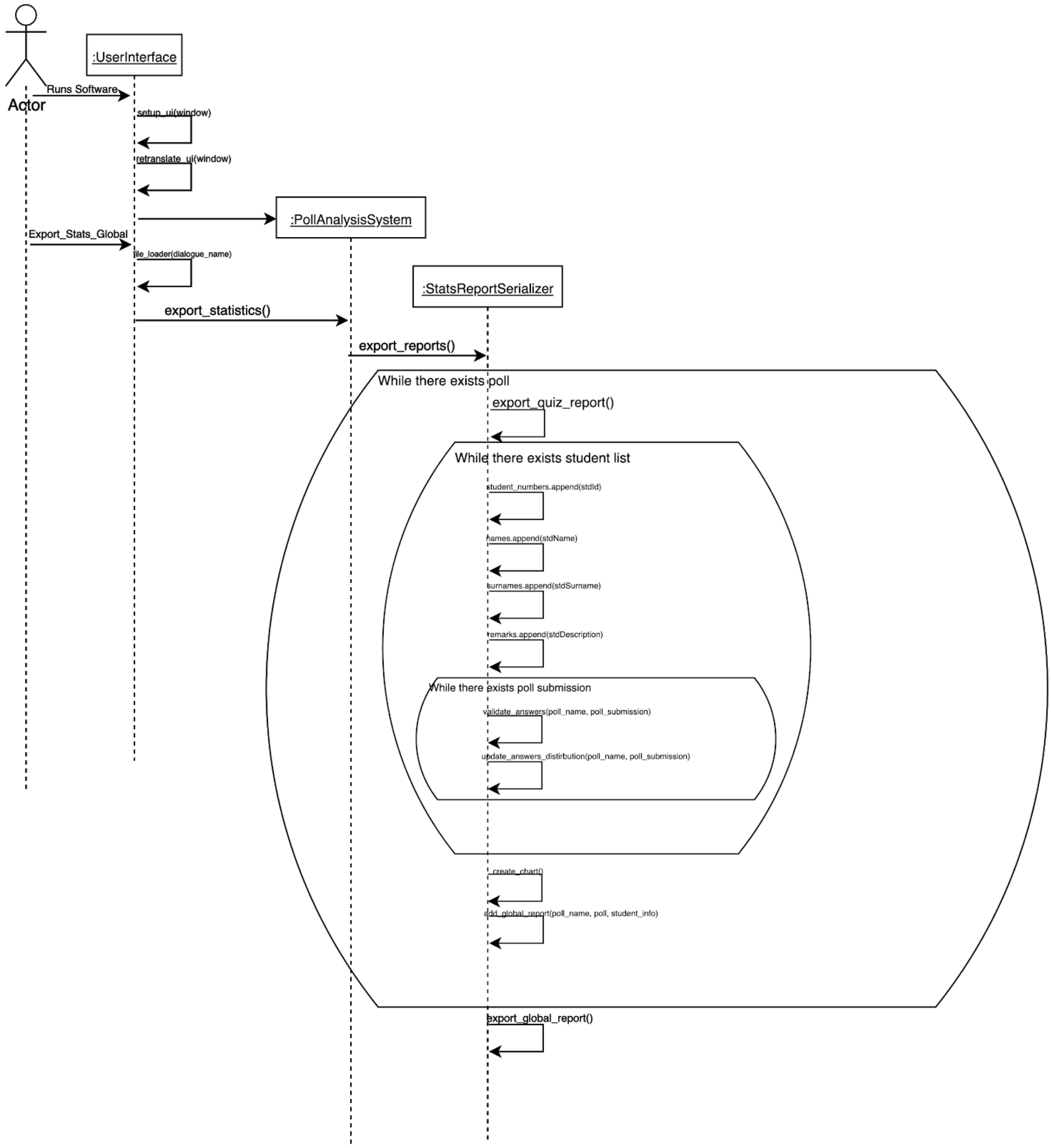
3.7.3 Load Poll Reports


Actor
:UserInterface
Runs Software
setup_ui(window)
retranslate_ui(window)
:PollAnalysisSystem
LoadPolls
file_loader(dialogue_name)
:PollParser
ReadPollReports(poll_reports_files)
While there exists a poll report
parse_poll_report(filename)
process_row(row)
process_questions_answers(questions_answers)
update_polls(poll_info)
AttendancePoll(poll_name, poll_date, poll_weekday)
:AttendancePoll
QuizPoll(poll_name, poll_date, poll_weekday)
:QuizPoll
PollSubmission(sub_datetime, poll, student)
:PollSubmission

3.7.4 Export Attendance Report


Actor
:UserInterface
Runs Software
setup_ui(window)
retranslate_ui(window)
:PollAnalysisSystem
ExportAttendance
file_loader(dialogue_name)
:AttendanceReportSerializer
export_attendance()
export_attendance_report()
get_db()
While there exists student list
While there exists registration
student_numbers.append(stdId)
names.append(stdName)
surnames.append(stdSurname)
remarks.append(stdDescription)

3.7.5 Export Statistics and Global Report


Actor
:UserInterface
Runs Software
setup_ui(window)
retranslate_ui(window)
:PollAnalysisSystem
Export_Stats_Global
file_loader(dialogue_name)
:StatsReportSerializer
export_statistics()
export_reports()
While there exists poll
export_quiz_report()
While there exists student list
student_numbers.append(stdId)
names.append(stdName)
surnames.append(stdSurname)
remarks.append(stdDescription)
While there exists poll submission
validate_answers(poll_name, poll_submission)
update_answers_distirbution(poll_name, poll_submission)
create_chart()
add_global_report(poll_name, poll, student_info)
export_global_report()

4 Glossary

Zoom : Zoom is a cloud-based video conferencing service you can use to virtually meet with others - either by video or audio-only or both, all while conducting live chats - ,it lets you record those sessions to view later, and allows you to create a single choice or multiple choice polling questions for your meetings.

Python: An object-oriented programming language that we will use to compose the tools we use in this project.

Interface: In computing, an interface is a shared boundary across which two or more separate components of a computer system exchange information. The exchange can be between software, computer hardware, peripheral devices, humans, and combinations of these.

User Interface: A user interface (UI) is the part of a machine, product, or device with which a person interacts. Usually visual and tactile in nature, a user interface is what an individual uses to make a machine or product do what he wants it to do. The user interface should be intuitive and simple to use.

Dataset: A data set is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question.

Database: A database is an organized collection of data, generally stored and accessed electronically from a computer system.

Operating System (OS): An Operating System (OS) is an interface between a computer user and computer hardware.

Attributes: Object specific properties are called, instance attributes. Think of it as the properties of instances.

Framework: A framework is a particular set of rules or ideas which you use in order to deal with problems or to decide what to do.

CSV: A comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. Each line of the file is a data record. Each record consists of one or more fields, separated by commas.

TXT: A TXT file is a standard text document that contains unformatted text. It is recognized by any text editing or word processing program and can also be processed by most other software programs.

Functional Requirements: In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs.

Non-functional Requirements: In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

Library: In programming, a library is a collection of precompiled routines that a program can use. The routines, sometimes called modules, are stored in object format. Libraries are particularly useful for storing frequently used routines because you do not need to explicitly link them to every program that uses them

GUI : The GUI(Graphical User Interface**)** user interface is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicators such as primary notation, instead of text-based user interfaces, typed command labels or text navigation.

Clustering: Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups

Class :In object-oriented programming, a class is an extensible program-code-template for creating objects, providing initial values for state (member variables) and implementations of behavior (member functions or methods).

Terminal: Terminals, also known as command lines or consoles, allow us to accomplish and automate tasks on a computer without the use of a graphical user interface.

Instance/Object: In a computer system, any time a new context is created based on some model(its class), we say that the model has been instantiated. In practice, this instance usually has a data structure in common with other instances, but the values

Object-Oriented Programming: Object-Oriented Programming(OOP), Procedural programming is about writing procedures or functions that perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions.

Parsing: To split a file or other input into pieces of data that can be easily stored or manipulated.