MARMARA
UNIVERSITY

CSE 3063 Object Oriented System Design

Doç. Dr. Murat Can Ganiz

Department of Computer Engineering

Marmara University

Fall 2020

Project 2

Group 20

150319067 - Elif Sünnetçi

150119933 - Nurhande Akyüz

150119686 - Diala Jubeh

150119900 - Emre Okul

150119746 - Alper Dokay

150119630 - Bilal Tan

150119640 - Ayberk Altuntabak

Requirement Analysis Design

Introduction

The development of technology in the world is progressing with great speed. Similarly, in education, we see that the areas of use of technology are expanding. With smart classrooms and labs, education becomes more interactive. While these developments were taking place, a worldwide pandemic negatively affected global education. Education came to a halt, especially in underdeveloped countries. Developed and developing countries are now trying to continue education online. The IT sector is working diligently to produce new tools for online education. There is an evaluation problem in online education because we are away from our old order. In this article, we present a new tool for academicians to analyze zoom pools in detail to support online education.

Purpose

Especially with the pandemic, there has been a big leap forward in online education. Different methods were tried to continue the education. Although different platforms were used, Zoom was able to highlight itself the most. With this program, while teaching in a video format, it can also be recorded. While academics provide education through this program, they use the pool, a feature provided by this program, to increase interaction with the student and test them.

Within the scope of this project, Csv files, which are the outputs of Zoom repositories, will be processed, stored and reports will be obtained in a way that will be useful for academics. With a detailed performance chart for each student, their progress can be followed. Through the interface to be developed, students will be able to read and analyze which exam and what result they got.

Glossary

Pandas: is an open source data analysis and manipulation tool, builded with Python programming language.

JIRA: is a software that is developed in order to help teams manage their work. It provides to practice agile methodologies such as scrum.

UML: short for unified modelling language. It is a standardised language which helps developers to keep track of the system. UML is a visualized language which represents the system with diagrams.

Use Case: is a set of cases or scenarios for using a system, tied together by a common user goal. It emphasizes the user's part of view and explains everything in the user's language.

Class: is a modeling tool provided by the programming language for a program or application to use in representing real world objects. Classes on this program are used to define data and user. Class Label's are chosen according to the objects they represent.

Instance: is a variation of an object created from a class. Users can create these object variants by using constructors.

Dataset: is usually designed for analysis rather than constantly updating from different users, so it represents the end of a data collection or a snapshot of a particular time.

User: is a person using a computer or network service. In this program, the academician who uploaded and analyzed the file she/he wanted in this program.

User interface: a platform for interaction between the user and the system; in order to analyze the installed data.

User Needs

A software requirement is a capability needed by the user to solve a problem or to achieve an objective. In other words, requirement is a software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documentation. Therefore, our program first receives data from the user. We analyze x, y, z, w columns from 4 columns here. We can view the detailed analysis made for the students in each row on the graphical interface. To give an example, We can reach the results of data such as the scores obtained by a student named Ahmet Öztürk from all the pools he attended and the percentage of participation in the lessons. As a result, the academician can interpret or grade each student individually.

System Features and Requirements

Functional Features and Requirements

- Users of the program can access the grades of the student they want.
- Data Sets are uploaded by the user.
- Program detects questions and finds the answer key by itself.
- Program generates detailed charts for user
- more feature
- more feature

Non-functional Requirements

- **Performance**

The program's import activities must be done in a short-time period in order to make a strong and powerful software for users. By executing the commanded activities, it should update the users in runtime with ongoing tasks. Student data should be imported and analyzed quickly. At the same time, the graphic creation unit should work fast. The output should be of the desired size and speed.

- **Reliability**

Since the data exported by the program will be in use of our algorithms, the program should work effectively. Users will get the benefit of different approaches with correctly implemented tools.

- **Recoverability**

Users will be able to have the files used by the program accessible and they can revert the changes or export new data files whenever they request from the program to do that.

- **Data Integrity**

Any dataset with the program's format will be available to import to the system. Users will have the privilege of importing any dataset over the program.

- **Interoperability**

The program will apply a tool-based orientation in control of users. Mechanisms will cooperate with both the data and the users so as to provide a diversity among program entities and users.

- **Security**

The datasets and outputs should be executed correctly and prevented loss of any data imported. Additionally, the program must be reachable for only the people having the program itself without any remote connection.

- **Maintainability**

If the program faces any problem in use, it should create a log for that and export it to the log file to be handled in next developments. Furthermore, the program should have the design to be extended with both current entities like mechanisms and possible new entities such as machine learning models or deep learning models.

User-interface

The program will be running as a console application. Hence, the scenario will be applied on a console screen. Any operation like data import, data export, record of any student or graph of the pool can be displayed.

Domain Model