HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF INFORMATION AND

COMMUNICATION TECHNOLOGY

--------------\*--------------

**GROUP PROJECT**

Course: Introduction to Software Engineering

**TOPIC: EXAM GENERATE SYSTEM**

Group: Something

Class code: 131594

Instructor: Trinh Thanh Trung

Group member:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Name | Student ID | Email | Class |
| 1 | Nguyễn Quang Minh | 20194444 | [minh.nq194444@sis.hust.edu.vn](mailto:minh.nq194444@sis.hust.edu.vn) | DS-AI 01 K64 |
| 2 | Hà Việt Hùng | 20194435 | [hung.hv194435@sis.hust.edu.vn](mailto:hung.hv194435@sis.hust.edu.vn) | DS-AI 01 K64 |
| 3 | Lê Thành Thắng | 20194451 | [thang.lt194451@sis.hust.edu.vn](mailto:thang.lt194451@sis.hust.edu.vn) | DS-AI 01 K64 |
| 4 | Lê Đức Anh | 20194416 | [anh.ld194416@sis.hust.edu.vn](mailto:anh.ld194416@sis.hust.edu.vn) | DS-AI 01 K64 |
| 5 | Nguyễn Thị Trang | 20194458 | [trang.nt194458@sis.hust.edu.vn](mailto:trang.nt194458@sis.hust.edu.vn) | DS-AI 01 K64 |

*Hà Nội, 07/2022*

Table of Contents

[PROJECT ASSIGNMENT 3](#_Toc110022729)

[1. Feasibility study 3](#_Toc110022730)

[Client Request 3](#_Toc110022731)

[Executive summary 4](#_Toc110022732)

[Application overview 5](#_Toc110022733)

[2. Software requirement analysis and design 5](#_Toc110022734)

[Functional requirement 6](#_Toc110022735)

[Software Interface 6](#_Toc110022736)

[Analysis model 7](#_Toc110022737)

[Entity relationship model 7](#_Toc110022738)

[Data flow diagrams 7](#_Toc110022739)

[Usercase diagram 9](#_Toc110022740)

[Database Design 10](#_Toc110022741)

[3. Building illustration program 10](#_Toc110022742)

[Library and tool 10](#_Toc110022743)

[Demo GUI 11](#_Toc110022744)

[Main Screen Interface 11](#_Toc110022745)

[Question manage 11](#_Toc110022746)

[Exam List Interface 13](#_Toc110022747)

[Exam manage interface 15](#_Toc110022748)

[Subjects manage Interface 17](#_Toc110022749)

[4. Software testing 19](#_Toc110022750)

[Modify question bank test case 19](#_Toc110022751)

[Modify subject test case 21](#_Toc110022752)

[Modify exam test case 21](#_Toc110022753)

[5. Installation and usage instructions 22](#_Toc110022754)

[Installation 23](#_Toc110022755)

[Intended Users and Software Scope 23](#_Toc110022756)

[Installation requirement 23](#_Toc110022757)

[Detailed Installation instruction 24](#_Toc110022758)

[Software Usage Guide 25](#_Toc110022759)

[CONCLUSION AND FUTURE WORK 26](#_Toc110022760)

# PROJECT ASSIGNMENT

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Student ID | Email | Work |
| Hà Việt Hùng | 20194435 | [hung.hv194444@sis.hust.edu.vn](mailto:hung.hv194444@sis.hust.edu.vn) | Exam GUI |
| Nguyễn Quang Minh | 20194444 | [minh.nq194444@sis.hust.edu.vn](mailto:minh.nq194444@sis.hust.edu.vn) | Question GUI |
| Nguyễn Thị Trang | 20194458 | [trang.nt194458@sis.hust.edu.vn](mailto:trang.nt194458@sis.hust.edu.vn) | Import and Export Data |
| Lê Thành Thắng | 20194451 | [thang.lt194451@sis.hust.edu.vn](mailto:thang.lt194451@sis.hust.edu.vn) | Database Design |
| Lê Đức Anh | 20194416 | [anh.ld194416@sis.hust.edu.vn](mailto:anh.ld194416@sis.hust.edu.vn) | Subject GUI |

# Feasibility study

## Client Request

Developing a system to create exam questions for teachers.

* The system can manage a question bank including multiple choice questions and constructed response questions.
* Each essay question has suggested answers.
* Each multiple-choice question has answer options and answers to that question. The number of options is unlimited.
* A multiple-choice question can also have multiple correct answers.
* Requires each question to belong to a certain subject.
* Each question can also be assigned a difficulty level, within one or more chapters of that subject.

The system must have functions:

* Manage questions by subject, chapter, difficulty
* When creating a test, teachers can create a test that mixes multiple-choice and essay questions or has essay sections and multiple-choice section separated.
* Questions are hand-picked or randomly generated according to pre-selected chapters and difficulty levels.
* Shuffle answer, question for the same exam
* Store exam information
* Export to file (txt, doc, pdf...)

## Executive summary

The proposed system for teachers of Hanoi University of Science and Technology. The customer is represented by Mr. Trinh Thanh Trung.

Our goal is to develop a question management system that can automatically generate exams for teachers at Hanoi University of Science and Technology. There are two categories of questions : multiple choice that can have multiple answers and written questions with hints for answers. The system will be able to separate them by subjects, chapters, and difficulties, allowing teachers to randomly create an exam with predefined difficulties and chapters, or hand- picked. For the same exams, questions and multiple-choice answers can be shuffled. Finally, teachers can save the exam and export to a file ( txt, doc or pdf ... ). We aim to automate the exam creations process.

## Application overview

The goal : The application should be able to automate the exams

creation process, either hand-picked or randomly using chosen subject, difficulties and chapters from a predefined question bank.

 Business goal : Through this system, we hope to create a more efficient way to store questions and previously created exams, and furthermore reduce the workload on teachers during exam weeks by replacing manual work with an automated process and user- friendly UI.

Current business process : Currently, even with the same subject, teachers of different classes still have to manually create their own exams.

 User roles : The system will have only one type of user with access to every feature

Interaction with other systems: The system will be built from ground-up using existing libraries. The exams generated by this system can be combined with an automatic testing and grading application for students in the future.

# Software requirement analysis and design

## Functional requirement

Major functions of the questions bank management system:

* Enable an user to view the list of every questions in the database
* Enable an user to view a question and it’s answer(s)
* Enable an user to view a question’s difficulty, subject and chapter
* Enable an user to add, remove or edit existing questions and answers

Major function of the exam generation system :

* Generate an exam either randomly or manually - View previously generated exams
* Edit an existing exam
* Remove an existing exam
* Export the exam into a docx file

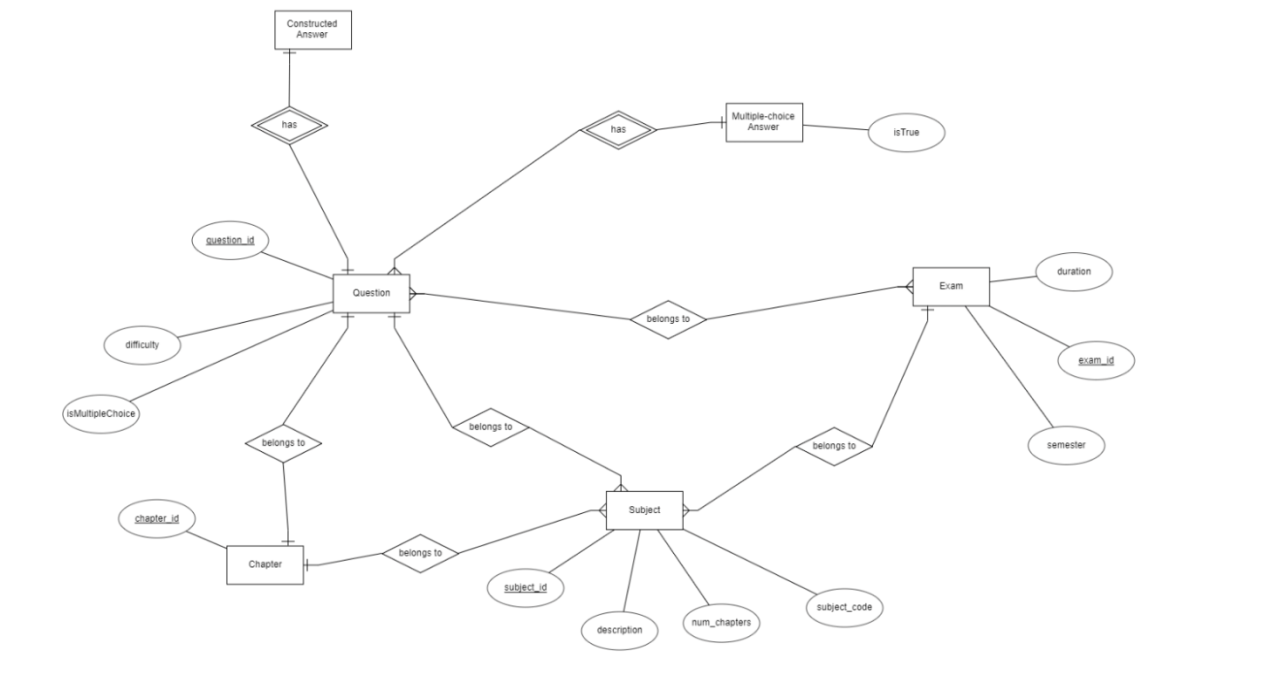
The system should be able to export a database or exam into a format that can be imported to another device

## Software Interface

|  |  |
| --- | --- |
| Operating systems | Window, MacOS |
| Database | To save questions and exams records, we use .CSV files stored on the client’s device |
| Language | To implement the project we choose the Python language |
| User Interface | For the user interface, we use the PyQt5 framework |
| Data manipulation | To save, edit and query data, we use the Pandas library |

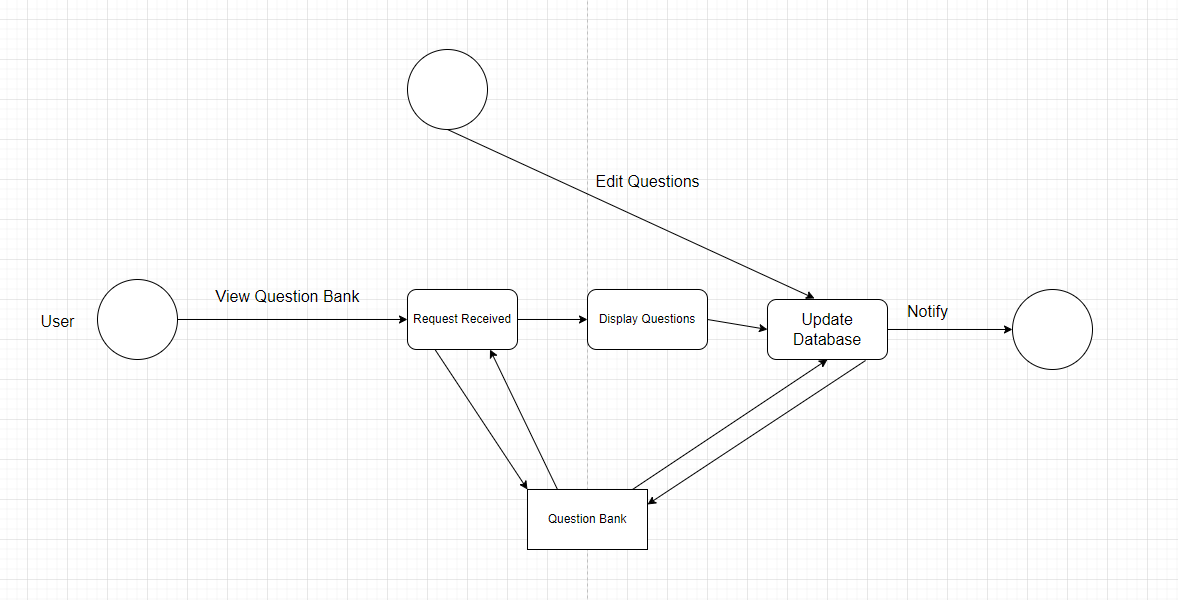
## Analysis model

### Entity relationship model

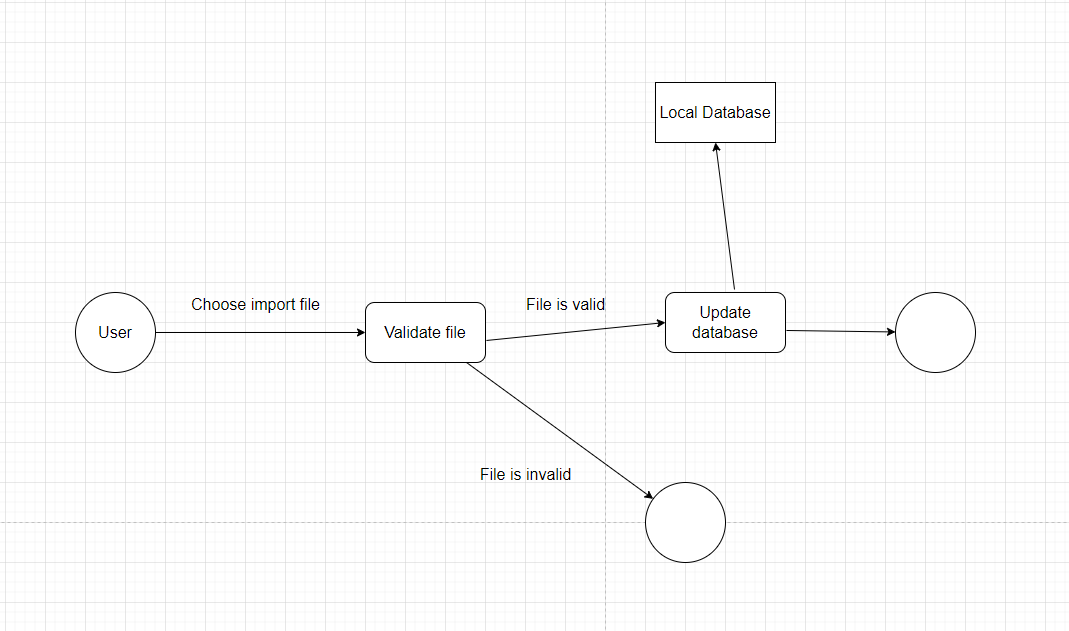


### Data flow diagrams

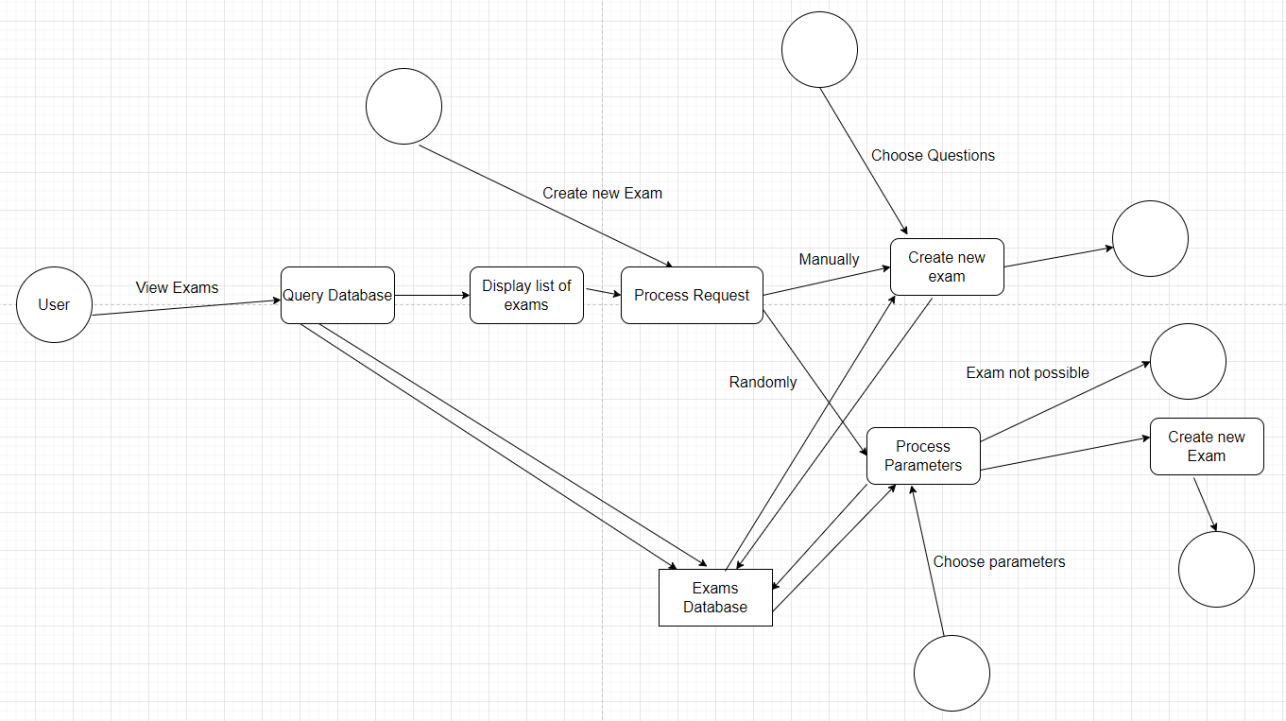
When viewing and editing question bank



When importing file



When viewing and generating exams



### Usercase diagram**Diagram Description automatically generated**Database Design

**Graphical user interface, text, application, chat or text message

Description automatically generated**

# Building illustration program

## Library and tool

PROGRAMMING LANGUAGE: PYTHON

**List of using tool**

|  |  |  |
| --- | --- | --- |
| **Purpose** | **Tool** | **URL** |
| **CODING IDE** | **PYCHARM COMMUNITY EDITION** | **https://www.jetbrains.com/pycharm/** |
| **Design GUI** | **QT DESIGNER** | **https://build-system.fman.io/qt-designer-download** |

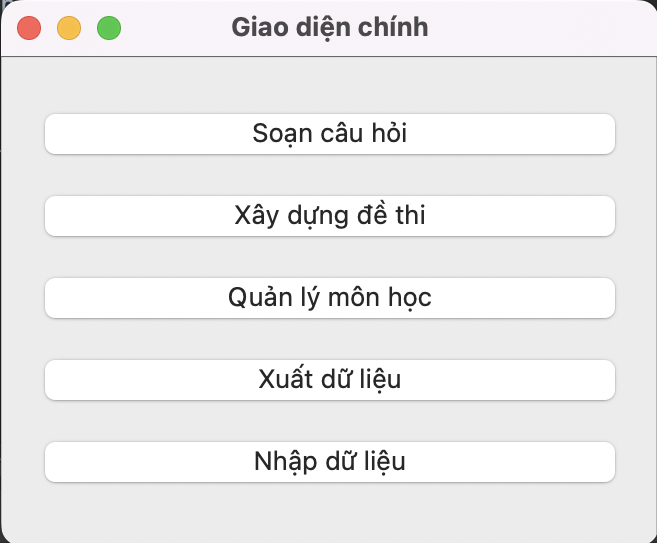
LIST OF USING LIBRARY

|  |  |
| --- | --- |
| Library | purpose |
| PyQt5 | Design GUI |
| Pandas | Read, write and retrieve data |

## Demo GUI

### Main Screen Interface

Main screen of the application



### Question manage

* Add, modify, and delete question of the question database.
* Add, modify, and delete answer or suggested answer from the answer database

Graphical user interface, text, application

Description automatically generated

Function:

* Switch between “Trắc nghiệm” and “Tự luận” radio button will change the question list regard to question type.
* When click “Thêm câu hỏi” label a pop up apprear to fill in the question.
* Edit “Chương” and “Độ khó” combo box change chapter and difficulty of question.
* When click “Thêm” label a pop up apprear to add answer option of multiple-choice question.
* When click “Chỉnh sửa” label a pop up apprear to modify the current clicked answer option of multiple-choice question.
* Click “Xoá” will delete current clicked answer option of multiple-choice question.
* Click “Xoá câu hỏi” button delete the current clicked question.
* Click “Lưu” button for saving changes.

### Exam List Interface

Show exam of subject

Graphical user interface, application

Description automatically generated

Function:

* Click “Tạo để thi mới” button to create new exam. This pop up appear for user to choose how to create new exam: Manually or automatically.

Graphical user interface, text, application

Description automatically generated

* Click the exam button (yellow background button) for open edit exam interface.

Create exam automatically interface: This interface shows when use choose to create exam automatically. Let user to automatically create exam based on type of question and based on difficulty (or chapters)

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Function:

* Choose “Dạng đề thi” combobox to choose type of question. Available option: “Trắc nghiệm”, “Tự luận”, “Trắc nghiệm và tự luận”.
* Choose “Thiết lập độ khó combobox to for how the exam will generated. Available option: “Độ khó” and “Độ bao phủ”.
* Fill in the number of questions user want to generate and click “Tạo đề thi”. After click, this frame will be closed and the exam manage interface will appear.

### Exam manage interface

This interface appear when user click generate exam manually, click the exam label from exam list interface, or after create exam automatically.

Purpose:

* Edit, add or delete question from exam.
* Edit exam information.
* Save, modify and delete exam.
* Export exam to text file.

Graphical user interface, text, application, email

Description automatically generated

Function:

* Edit question score in “Điểm cho câu hỏi” Spinbox.
* Click “Thêm vào đề thi” button to add the current clicked question of question list to the exam.
* Click “Xoá câu hỏi ra khỏi đề thi” label to delete the curent clicked question in exam question list
* Click Up and down arrow button to move the current clicked exam question up and down.
* Fill the exam infomation (Time, Year, Semester)
* Click “Xuất ra file” label to save exam to file (txt, rtf, …)
* Click “Xáo trộn các câu hỏi” to shuffle the question in exam.
* Click “Lưu đề thi” button to save exam, “Xoá đề thi” to delete exam.

### Subjects manage Interface

Add, modify, or delete subject

Graphical user interface, text, application

Description automatically generated

Function:

* Edit subject infomation
  + Modify “Tên môn học” text edit to modify subject name.
  + Modify “Mã học phần” text edit to modify subject code.
  + Modify “Số chương” spin box to modify number of chapters.
  + Modify “Giới thiệu” text edit to modify description.
* Click “Lưu” button to save subject information.
* Click “Thêm môn học” to open add subject pop up.
* Click “Xoá môn học” button to delete current subject.

Graphical user interface, application

Description automatically generated

# Software testing

## Modify question bank test case

* Test cases for Add Questions : require user to input the question, question type, chapter, difficulty
* The subject must be chosen beforehand and the user can only choose chapters and difficulty that exists within that subject

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Question** | **Type** | **Chapter** | **Difficulty** | **Result** |
| 1 | …… | null | 1 | 1 | Display Error Message |
| 2 | null | Tự luận | 1 | 1 | Display Error Message |
| 3 | …… | Tự luận | 2 | 3 | Accepted |
| 4 | ……. | Trắc nghiệm | 3 | 2 | Accepted |

Test cases for Add Answers : require the user to choose the question, and provide an answer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Question Type** | **Answer** | **isCorrect** | **Result** |
| 1 | null | …… | null | Display Error Message |
| 2 | Tự luận | …… | null | Accepted |
| 3 | Trắc nghiệm | …… | True | Accepted |
| 4 | Trắc nghiệm | null | True | Display Error Message |

## Modify subject test case

* The Add subject use case require the user to input the subject’s name, code, number of chapters and description

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | **Name** | **Code** | **Number of chapters** | **Description** | **Result** |
| 1 | Introduction to Software Engineering | IT3180 | 4 | ….. | New subject is added |
| 2 | null | IT3180 | 4 | …. | Display Error Message |
| 3 | Introduction to Software Engineering | null | 4 | …. | Display Error Message |
| 4 | Introduction to Software Engineering | IT3180 | null | …. | Display Error Message |
| 5 | Introduction to Software Engineering | IT3180 | 4 | null | Display Error Message |

## Modify exam test case

* Create exam manually use case require the user to input:
  + Questions to be included in the exam
  + The exam’s duration (in minutes)
  + The exams’ semester

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Questions List** | **Duration** | **Semester** | **Result** |
| 1 | …. | 50 | 20212 | Accepted |
| 2 | … | s | 20211 | Display Error Message |
| 3 | … | 50 | s | Display Error Message |
| 4 | … | null | 20211 | Display Error Message |
| 5 | … | 50 | Null | Display Error Message |

# Installation and usage instructions

## Installation

Graphical user interface, application, Teams

Description automatically generated

## Intended Users and Software Scope

* Intended users: Teacher.
* Software Scope: Question database management, Exam database management, generate exam based on question database.

## Installation requirement

Hardware requirement: None

Software requirement:

* Version control (Git)
* Python3
* Python3 library
* Python package manager (PIP)

## Detailed Installation instruction

The software is programmed using Python3 language, so it is necesssary to have python3 environment.

Python3

Check python version in your environment using the command:

python --version

If there is some error or python version is <= 3.x.x then a new version of python is required.

Python3 can be downloaded from the homepage

<https://www.python.org/downloads/>

PIP

PIP is already installed if you download python3 from homepage(above). Incase you dont have, you can type the following command to install

MacOS and Linux user:

python -m ensurepip --upgrade

Window user:

py -m ensurepip --upgrade

Installation guide can be found here:

<https://pip.pypa.io/en/stable/installation/>

Git Version Control

Installation guide:

<https://git-scm.com/downloads>

When all requirements are satisfied, user can go back to 7.1 to install.

## Software Usage Guide

The software has 3 main functions: question manage, exam manage, and question manage. They can be access when click from the main screen. The button functions are clearly defined in Section 5.3

# CONCLUSION AND FUTURE WORK

Our software has satisfied most of client’s requirements, which include:

* Storing questions
* Manage questions based on subject, chapter, and question difficulty
* Generate exam automatically or manually based on chapter or based on difficulty
* Shuffle question in an exam
* Store exam information
* Export exam to a file

However, due to time constraint, some of the functions are simplified or cannot work as the way we intend. For example, the current import function will overwrite data of the user. Although this solves the data conflicts problem (happens when 2 users edit the same question), it will delete all the question of the 2nd user, which is not what they might want.

In the future, we will include online server so that different user can access the same exam and same question, which solve the data conflict problem. We will keep the offline database and solve the data conflict problem by notify the user which question is duplicated and modified and provide user option to handle the situation.