

**College of Engineering**

**COMP 491 – Computer Engineering Design**

**Project Proposal**

**KUPEP (Protecting Exam Platform)**

**Spring 2021**

**Participant Information:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **ID** | **Email** | **Phone** |
| Umut Günçer | 60598 | uguncer16@ku.edu.tr | +90 530 944 84 94 |
| Yiğit Çırak | 49730 | ycirak14@ku.edu.tr | +90 543 879 99 93 |
| Batuhan Acar | 60186 | Bacar16@ku.edu.tr | +90 532 301 48 62 |
| Taluhan Öneş | 60035 | Tones16@ku.edu.tr | +90 537 717 74 12 |

**Project Advisor:**

Serkan Çil

**Abstract**

“An Exam Software is a carrier for easy online assessment that helps in general academic evaluation or exploring specific abilities and characteristics for talent selection” [1]. This exam software is widely used in various of environments, including universities, public and private selection authorities, businesses, and a variety of home-testing platforms. Exam software is required in such environments because it provides a high level of security framework, making it easier to ensure that cheating does not occur. When compared to a handwritten exam, it will take less time. Exam software makes it easier to prepare for an exam. Also, by having instant results, one will be able to see the performance analysis. By connecting to your computer, you can access the exam from any location. Exam solutions are divided into two categories: online and offline. Online tests are not restricted to a specific geographical area and can be completed from anywhere on the globe. Offline tests are similar to online exams in that they don't require internet access. In an offline test, a local intranet server or a LAN cable is used.

Data storage, detailed exam records, status, and statistics are all functions that both online and offline test solutions have. Exams in computer labs at Koc University are always software-based, such as Excel, MATLAB, Java, C, and so on. There is no software-based exam software available in the computer lab.

Our goal is to create an offline-software based lab exam solution for use in computer labs that is simple to use for both examiners and exam takers while safeguarding exam content and detecting and preventing cheating. Because our university's current lab exam solution is a manual system rather than a software solution, it has some drawbacks, such as verbal communication, the possibility of cheating, and exam takers' doubts about whether or not they have successfully submitted their test results. Exam takers can also access sites that are part of the local network, such as Kusis and KuBlackBoard.

We want to develop a solution that meets all of the requirements for a secure and user-friendly offline-software-based lab exam. In most cases, the examiner will begin by turning on the main computer that will be used to administer the exam. The exam takers' clients can automatically connect to the main computer once it has started. The examiner will then upload the exam content and establish exam rules (yes/no USBs, yes/no internet connection, exam time, help button availability, type of files an exam taker can submit, and so on). After that, the exam will be sent to the test takers. Exam takers can now access the exam content and see how much time is left. If an examiner needs to make an announcement, he can send a public message to all students, and if he needs to warn a student, he can send a private message to that student. Exam takers can request assistance by pressing the help button on their program. The exam-taker or assistants can then approach the student or allow the student to communicate through a private message box. The examiner will be able to determine whether or not the students are connected. If an exam taker's computer shuts down by accident, the examiner will be able to see who has disconnected and will resend the exam once the exam taker has restarted the computer. The student will not be able to press the help button at that time because the computer is closed. Once an exam taker has completed his work, he will be able to submit to the examiner the files in his working directory (files specific to the program he is using: Java, MATLAB, Excel, C, etc.). The examiner can keep track of who has turned in their work. Finally, the examiner will be able to transfer all of the submitted files, and the exam will be completed. The admin of the IT department will have access to the system's logs.

**Table of contents**

[Section 1 Introduction 4](#_Toc66229356)

[1.1 Concept 4](#_Toc66229357)

[1.2 Objectives 4](#_Toc66229358)

[1.3 Background 4](#_Toc66229359)

[Section 2 S/T methodology and associated work plan 5](#_Toc66229360)

[2.1 Methodology 5](#_Toc66229361)

[2.2 Work Package Descriptions 6](#_Toc66229362)

[2.3 Demonstration 10](#_Toc66229363)

[2.4 Impact 10](#_Toc66229364)

[2.5 Risk analysis 11](#_Toc66229365)

[2.6 Gantt Chart 12](#_Toc66229366)

[Section 3 Economical and Ethical Issues 13](#_Toc66229367)

[Section 4 References 14](#_Toc66229368)

1. Introduction
   1. Concept

We will be developing a lab exam software that will be used by both the examiners and the exam takers. Our solution will replace the current solution that has several disadvantages and our focus is to eliminate these disadvantages and provide a superior design solution. In the first phase we will analyse the project requirements and also document the current system’s disadvantages. Running parallel to this phase is the technology search phase where we will be searching for the appropriate technology on how to disable and enable the internet connection during the exam. Our third phase includes user interface developing both for the examiner and for the exam taker considering the ups and downs of the design aspects in the similar applications. Later we will move on to the software development phase, where the exam taker client will be developed after the examiner program has been developed. The fifth phase will be testing phase, first we will develop test cases according to the functional analysis, and once the software is ready for testing, we will conduct the tests and document the results. According to the bugs found in this phase we will bug fix and finalize the software ready for deploying. The last phase will be the deployment phase where the program will be deployed in the lab environments, tests will be conducted over again, and according to the test results final bug fixings will be made and the program will be ready to user.

* 1. Objectives

Our main objective is to implement a system to configure and control a computer lab that enables computer-based assessments while protecting the exam content, detecting and preventing cheating.

Basic Assumptions:

* All test takers take the exam at the same time in a physical computer lab room.
* Any desktop software can be used for assessments (e.g. Matlab, MS Excel etc).
* Internet access and communication between test-taker devices should disabled when the exam starts.
* Test taker should be able to save and upload the test results to a designated protected location during the exam only.
* Internet access and communication between test-taker devices should enabled when the exam ends.
  1. Background

Although there are several online and offline exam solutions on the market, there is no offline-software based solution available. That is why our university is using a non-software solution for lab exams. It is based on folder sharing and mostly verbal communication. The main disadvantages of the current situations are:

* All communication is done verbally, causing the exam takers to be disrupted.
* USB s are not disabled, which is a security flaw.
* Once the exam takers PC shutdown unexpectedly during the exam he would need assistance to restart it, but sometimes it is not possible to reach the examiner.
* Exam takers cannot be sure if they have submitted their work correctly, after the exam they form a line in front of the examiner and every exam taker confirms the submission one by one.
* Submission is confusing, exam takers are not sure which files to submit.

1. S/T methodology and associated work plan

* 1. Methodology

Since the requirements for our project are subject to change and it is not possible to divide the requirements into small deliverable phases, we have decided to use the waterfall methodology. The main phases for our project will be

1. Project requirements: In this phase we will gather and document the requirements for our project both from the perspective of the exam taker and examiner and admin with the opinions from the test takers and the examiners. We will also document the current system’s disadvantages.
2. Technology search: We will search for the appropriate technology to be used in our project specially to disable and enable the internet connection during the exam. This technology will be used in our software development phase.
3. User interface developing: An easy to use software and a useful design is one of our top priorities. We will first examine the user interface for similar applications that are available. Then we will design user interfaces both for the examiner and for the exam taker considering the ups and downs of the design aspects in the similar applications. By the end of this phase the UI will be documented.
4. Software development: The examiner side of the project will be developed first, since this will be the main program that the clients will connect to. The client and the server (examiner) will communicate through the Java network programming. After the examiner side is finished, we will develop the exam taker’s client program. Java programming language will be used both for the examiner and the exam taker side. We will use the network programming skills we have learned in COMP302 and COMP416.
5. Testing: We will first develop the test cases that will be used in the test phase. These test cases will cover the scenarios on how to test the program. Once the software development phase is finished, we will conduct the tests and prepare a test results document. After that we will bug fix our program accordingly and finalizing it to be ready for deployment. The testing will be done on local computers, simulating the lab environment. We are planning use virtual box. For exam-takers we will use 2 virtual boxes, for examiner we will use our local computer , for proxy we will use one virtual box .
6. Deploying: The software will be deployed in the labs. (School is closed due to pandemic will be revised later). After the deployment we will re-make all the tests in the lab environment. Lastly, we will make the last bug fixing and the program will be ready to use.
   1. Work Package Descriptions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Work package number | **1** | Start date or starting event: | | **Week 1** | |
| Work package title | **Project requirements** | | | | |
| Participant number | 1 | 2 | 3 | 4 | 5 |
| Participant name | Batuhan Acar | Umut Günçer | Yiğit Çırak | Taluhan Öneş |  |
| Weeks per participant | **1** | **1** | **1** | **1** |  |

|  |
| --- |
| **Objectives**   * Provide a well-documented requirements document that is subject to minor changes only. * Document current system disadvantages and the new system’s requirements. |
| **Description of work**  **T1.1 (w1)** DocumentCurrent system disadvantages  We will prepare a report that will document the disadvantages of the current system.  **T1.2 (w1)** Requirements for the new system – Examiner  The requirements for the examiner part of the program will be documented. Examiner will be able to manage the exam.  **T1.3 (w1)** Requirements for the new system – Exam Taker  The requirements for the exam taker part of the program will be documented.  **T1.4 (w1)** Requirements for the new system – Admin  The requirements for the admin part of the program will be documented. Admin will be able to retrieve program logs.  **T1.5 (w1)** Finalize the requirements document  All the documents that are produced in the first 4 tasks of this section will be merged into a single functional requirements document. |
| **Deliverables**  **D1.1** (w1) Functional Requirements document |
| **Milestones**  **M1.1** (w1) Requirement document is ready for UI Development |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Work package number | **2** | Start date or starting event: | | **Week 1** | |
| Work package title | **Technology Search** | | | | |
| Participant number | 1 | 2 | 3 | 4 | 5 |
| Participant name | Umut Günçer | Batuhan Acar |  |  |  |
| Weeks per participant | **2** | **2** |  |  |  |

|  |
| --- |
| **Objectives**   * Search for the appropriate technology (internet enabling and disabling) |
| **Description of work**  **T2.1 (w2)** Search for the appropriate technology (internet enabling and disabling)  We will research for how to disable and enable the internet connection during the exam. This knowledge will be used in the software development phase. |
| **Deliverables**  **D2.1 (**w2) Technology Search document |
| **Milestones**  **M2.1** (w1) Technology Search document is ready for Software Development |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Work package number | **3** | Start date or starting event: | | **Week 2** | |
| Work package title | **User Interface Developing** | | | | |
| Participant number | 1 | 2 | 3 | 4 | 5 |
| Participant name | Taluhan Öneş | Yiğit Çırak |  |  |  |
| Weeks per participant | **1** | **1** |  |  |  |

|  |
| --- |
| **Objectives**   * Develop a user-friendly interface both for the exam taker and the examiner. |
| **Description of work**  **T3.1 (w2)** Examine the user interface for similar applications  We will examine the user interface for similar applications that are available.  **T3.2 (w2)** Design user interface for the examiner  We will design a UI for the examiner part of the software considering the ups and downs of the design aspects in the similar applications.  **T3.3 (w2)** Design user interface for the exam taker  We will design a UI for the exam taker part of the software taking into account the ups and downs of the design aspects in the similar applications.  **T3.4 (w2)** Finalize UI Design document  The design for the exam taker and the examiner will be merged and finalized into one final document |
| **Deliverables**  **D3.1** (w2) User Interface Design |
| **Milestones**  **M3.1** (w2) User Interface Design is ready for Software Development |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Work package number | **4** | Start date or starting event: | | **Week 3** | |
| Work package title | **Software Development** | | | | |
| Participant number | 1 | 2 | 3 | 4 | 5 |
| Participant name | Batuhan Acar | Umut Günçer | Yiğit Çırak | Taluhan Öneş |  |
| Weeks per participant | **3** | **3** | **3** | **3** |  |

|  |
| --- |
| **Objectives**   * Develop a bug-free software that matches the requirements using Java and network programming. * Github should be extensively used. |
| **Description of work**  **T4.1 (w4)** Development of the examiner  The examiner side of the project will be developed first, since this is will be the main program that the clients will connect to. The client and the server (examiner) will communicate through the Java network programming. Our program will use a MYSQL database to store the logs. Admins can log in the database and see the logs.  **T4.2 (w5)** Development of the exam taker  The exam taker’s client program will be developed.  **T4.3** **(w6)** Finalize the programming  Make final developments both for the examiner and the exam takers side. Make sure all the functions have been covered and developed. |
| **Deliverables**  **D4.1** **(w6)** All the program code will be committed to Github by the end of w6 |
| **Milestones**  **M4.1** **(w6)** Program is ready for testing |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Work package number | **5** | Start date or starting event: | | **Week 6** | |
| Work package title | **Testing & Bug fixing** | | | | |
| Participant number | 1 | 2 | 3 | 4 | 5 |
| Participant name | Batuhan Acar | Umut Günçer | Yiğit Çırak | Taluhan Öneş |  |
| Weeks per participant | **2** | **2** | **2** | **2** |  |

|  |
| --- |
| **Objectives**  **WP5**   * Prepare test cases, test and bugfix the software |
| **Description of work**  **T5.1 (w4)** Prepare test cases  Test cases will be developed after the functional requirements are ready. We will document the scenarios on how to test the program and make sure the program is running correctly.  **T5.2 (w6)** Testing  All the test cases that have been created in task 5.1 will be tested on the first version of the program. The results of the test will be documented.  **T5.3 (w7)** Bug fixing test taker  Bug fixing for the exam taker side will be done according to the test results.  **T5.4 (w7)** Bug fixing examiner  Bug fixing for the examiner side will be done according to the test results.  **T5.6** **(w8)** Finalize the programming  Make final developments both for the examiner and the exam takers side. Make sure all the test result issues have been corrected. |
| **Deliverables**  **D5.1 (w4)** Test cases document  **D5.2 (w6)** Test results  **D5.3 (w8)** Final program code will be committed to Github by the end of w8 |
| **Milestones**  **M5.1** **(w4)** Test cases ready for testing  **M5.2** **(w6)** Test results ready for bug fixing  **M5.3** **(w8)** Program is ready for deploying |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Work package number | **6** | Start date or starting event: | | **Week 8** | |
| Work package title | **Deployment** | | | | |
| Participant number | 1 | 2 | 3 | 4 | 5 |
| Participant name | Batuhan Acar | Umut Günçer | Yiğit Çırak | Taluhan Öneş |  |
| Weeks per participant | **1** | **1** | **1** | **1** |  |

|  |
| --- |
| **Objectives**   * Deploy the software to the lab * Make sure the software is running correctly in the lab environment |
| **Description of work**  **T6.1 (w9)** Deploy the project  Deploy the project in the lab with the help of IT.  **T6.2 (w9)** Test the project  Test the project according to the test cases, this time in the lab environment and document the bugs found.  **T6.3 (w10)** Final bug fixing  Make the final bug fixings if necessary. |
| **Deliverables**  **D6.1 (w9)** Bug and fixes report for the lab environment |
| **Milestones**  **M6.1 (w9)** Test report is ready for final bug fixing  **M6.2 (w10)** Program is ready to use. All the final code is committed to Github. |

* 1. Demonstration

For performance analysis, we will control the each feature we are planning to implement and check whether they are working in a correct way or not. We will test our program from both the perspectives of examiners and exam-takers. For the perspective of examiners, we will check whether uploading exam content, closing or opening internet connection, preventing usage of USB, help alert, submission directory restriction, seeing IP addresses, connection status of exam-takers, public message sending, private message sending, sending exam to clients, starting the exams, the timer, are working without any bugs or flaws. If an examiner wants to let USB or cancel help feature during the exam, the changes, examiner will make in the exam settings, will be seen in the public message and every test taker will see it.

For the perspective of exam-taker, we will check whether the exam content that the examiner has sent, is coming up to exam-taker’s program. Moreover, pressing help button, submission restriction which will not accept the unrelated directory files, usage of USB, entrance to the websites in both external and internal network and the timer will be controlled. We will also demonstrate the scenario when a PC shutdowns unexpectedly.

In the final demonstration, our main purpose is to demonstrate a fully working program that provides all the features that we have defined. We will conduct a test from start to finish, including roles of the examiner and roles of the test taker. Since this program will work in only computer labs and because of pandemic the school is closed now, we hope that we will have a chance to show our final project in the university labs.

* 1. Impact

At the end of this project, we will have developed an offline-software platform for computer labs in Koç University. This new system will provide some convenience for both the examiners and the exam-takers when it is compared to the current system.

From an examiner’s perspective, this new platform will provide exam settings that the examiner can adjust some features before or during the exam. They will be able to control the submission directory, see whether exam-takers are connected to main computer, cancel usb usage and will restrict the use of internal and external sites. Moreover, they will be able to communicate with exam-takers from public and private message box. This way, there will be less communicational exam environment, and no one will be bothered during the exam. Examiner and exam-taker can solve problems by using private message box or examiner can make announce by using public message box. If an exam-taker seems disconnected in the main panel (in examiner’s computer), examiners will be able to see it and they can go next to exam-taker and see whether exam-taker has a problem with his computer or not. Also, examiner will be able to see who has submitted their work. In the current system, all exam-takers are coming and checking whether they have submitted correctly. So, examiner will not have to wait after exam and show all the works they have done. Moreover, as examiner will choose which directory exam-taker should submit, examiners will only open the submitted work instead of opening the whole project and finding the directory inside the work package as our current system is continuing like this. In the current system, teachers are showing how much time left on the school board by manually writing on the class board and only changing it when limited time has left like 5min, 15min, 30 min and they are announcing to whole class by shouting which is so distracting. So, teachers will just enter exam duration from our program in the settings before starting the exam and all exam-takers will see their exam duration. Examiner will decide whether they will make an online or offline exam by pressing the internet button in the exam settings. Some lab exam and quizzes are made by using internet. So, exam-takers maybe need to access their mail accounts, KuBlackboard or Kusis to get necessary materials for their class.

From an exam-taker’s perspective, this new platform will help students during the exam. There will be a help button and by this way they can call assistants or examiners to their side or if the examiner lets student use private message, they will be able to communicate from that platform without bothering anyone in the exam time. Moreover, Exam-takers will be certain about their submission. The biggest problem in our current exam system is that students are making a line in front of examiner’s computer and checking whether their projects are submitted or not. Because of the exam time stress, exam-takers are uploading whole project and thinks that examiners or assistants will extract the necessary parts and they will only be graded from that part. However, by sending the whole work, exam-takers get less grades. For preventing this submission problem, we will eliminate the other unnecessary work and will not let upload the whole project. This program will also prevent cheating by cancelling the usage of usb and sites, so that students will not be able to enter their mail accounts, kusis and ku blackboard accounts or make a usb delivery between their friends.

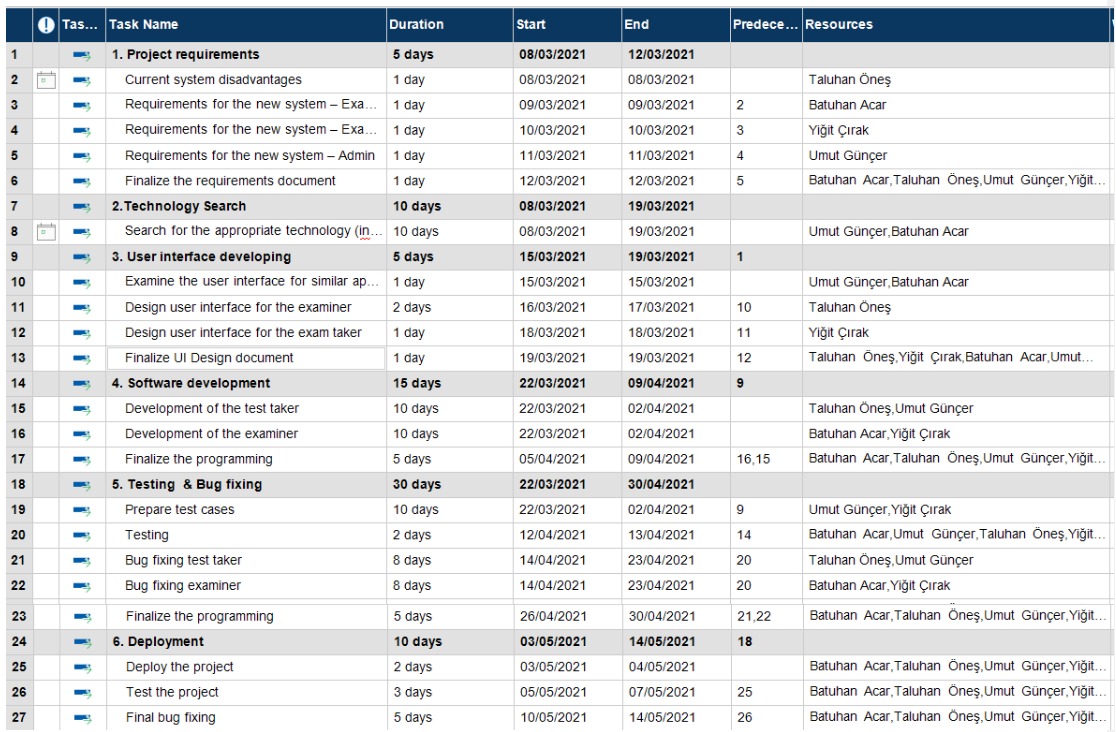
* 1. Risk analysis

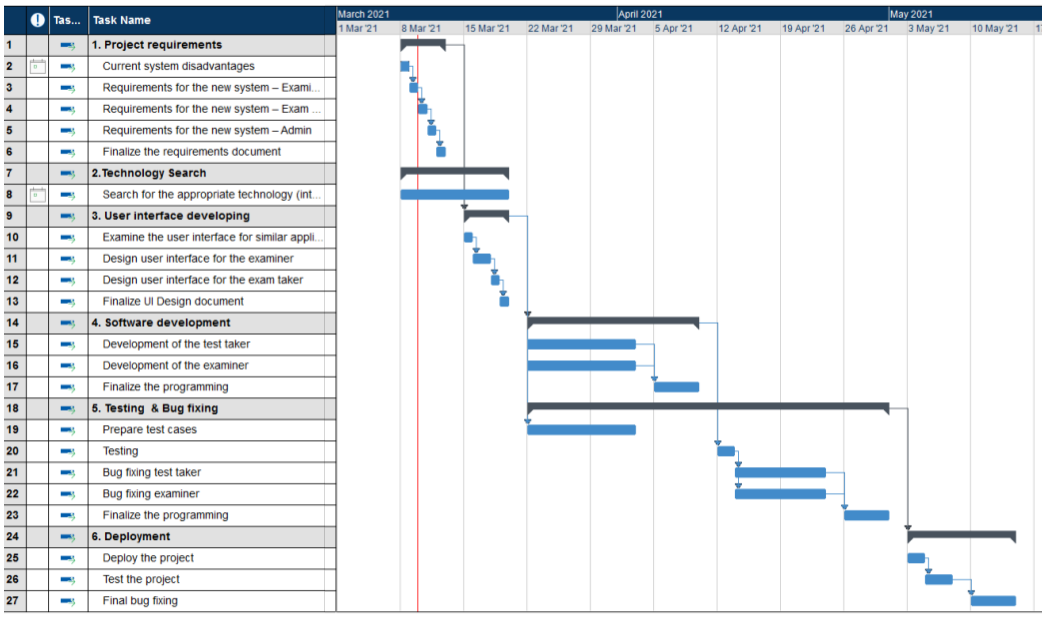
One risk is that instead of using help button and private message box for warning students or help exam-takers for technical or other issues, examiners or assistants can give some tips or solutions of the questions from private message platform during the exam. This is not ethical. But, all the examiners and exam-takers' actions will be logged. So, if any private message between examiner and exam-taker provides a benefit for one of them, Admin in IT department of the university will be able to see the actions and all messages.

Another risk is that sometimes in computer labs, computers are suddenly shut down. So, when an exam-taker seems disconnected in examiner’s computer, they will be seen red and disconnected in examiner's panel. When exam-takers again open the program, they will be seen connected and green and will be able to continue the exam from where they left off.

Lastly, if an exam-taker tries to kill the process from task manager, examiner will lose the connection with exam-taker and exam-taker will be able to use USB, internet or other stuff. So, to avoid this situation, IT department of the university should be called before the exam and admin will block the usage of task manager.

* 1. Gantt Chart





1. Economical and Ethical Issues

**Economical Issues:**

The product will be used in computer labs of the university and since it does not consist of any hardware part, no capital is needed. In other words, this project depends neither on the financial situation of developers nor the users.

**Realistic constraints**:

We can explore this in four different ways: Reliability, Usability, Learnability, Availability.

Usability: Our system has been designed to be straightforward and user friendly. The user interface consists of push buttons, time specifying parts, and LEDs. Our program's language is English. Therefore, the exam-takers and examiners are required to have pre-intermediate or higher English knowledge.

Reliability: All actions will be logged. Admin will be able to check every action by looking to database.

Availability: Our system depends on main examiner computer. When the examiner logs in to program, other clients will be able to connect to the system. The system will be able to run without internet connection.

Learnability: Computer labs play significant role in the education process not only for computer engineering department, but also for every department that are planning to do a computer exam. Therefore, this program will be user-friendly and straightforward that non-engineer students and teachers will execute the computer lab exam process smoothly.

**Ethical Issues:**

We decided to follow the following rules of this code during our project [2]:

3.08. Ensure that specifications for software on which they work have been well documented, satisfy the users requirements and have the appropriate approvals.

3.10. Ensure adequate testing, debugging, and review of software and related documents on which they work.

3.12. Work to develop software and related documents that respect the privacy of those who will be affected by that software.

5.11. Not ask a software engineer to do anything inconsistent with this Code.

By following these rules, all examiners and exam-takers' actions will only be seen by the admin checking up the logs and the database. As we figured out the shortcomings in the current system, this program will provide all the new features we have decided for both examiners and exam-takers. If any errors, bugs or updates are encountered, IT department of the university will only have right to fix it. Our project advisor, IT department or the developers of this project will not disrupt the code for the benefit of themselves.

1. References

[1] The Best 7 Free and Open Source Exam Software Solutions, Retrieved March 09, 2021, from

https://www.goodfirms.co/blog/best-free-open-source-exam-software-solutions

[2] Code of Ethics, IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices, Retrieved March 09, 2021, from

https://www.computer.org/education/code-of-ethics