**DOKUZ EYLUL UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

QUIZ HUB

**CME-2210**

**OBJECT ORIENTED ANALYSIS AND DESIGN**

**PROJECT -1**

**FINAL REPORT**

**BY**

**Hazar ÖZYAĞCI – 2017510090**

**Ayberk KILIÇASLAN – 2017510053**

**Arif MERTASLAN – 2017510063**

**Lecturers**

**Assoc.Prof.Dr. Semih UTKU**

**Res.Asst. Okan ÖZTÜRKMENOĞLU**

**Res.Asst. Yunus DOĞAN**

**IZMIR**

**27.05.2020**

**IZMIR**

**25.05.2020**

GHOSTRENTAL Car Service

**Table of Contents**

*TC*

1. Introduction 3
   1. [What the Problem is](#_TOC_250015) 3
   2. [Goals for the Project 3](#_TOC_250014)
   3. [Motivation for the Project 4](#_TOC_250012)
   4. [Process Flow Preview 4](#_TOC_250011)
2. Analysis and Design 5
   1. [Plan for Requirements Engineering 5 - 7](#_TOC_250010)
   2. [Functional Requirements 8](#_TOC_250009) - 10
   3. [Non-Functional Requirements 11](#_TOC_250008) - 13
   4. [Use Cases 14](#_TOC_250007) - 16
   5. [Models 16](#_TOC_250006) - 23
3. Project Plan 24
   1. Task Description 24 - 25
   2. [Task Assignment 26](#_TOC_250005)
   3. [Deliverables and Milestones 27](#_TOC_250004)
   4. [Project Schedule 27](#_TOC_250003) - 28
4. Testing 29
   1. [Features to be Tested 29](#_TOC_250002)
   2. [Test Cases 29](#_TOC_250001)
   3. [Testing Schedule 30](#_TOC_250000)

# Conclusion 31

* 1. The Problem and Solution 31
  2. The Team and the SE Process 32
  3. Engagement of Umbrella Activities 32 - 33
  4. The Stakeholder’s that Benefited 33
  5. The Organization’s Benefits 33

# User Manual 34

* 1. Software Description 34
  2. How to use the Software 34 - 35
  3. Troubleshooting Common Problems 35 - 36

**Introduction**

*1.*

# What the Problem is

Recently, in the world, because of the incidents distance learning is getting more important day by day. Unfortunately, there is not any helpful quiz application for the distance learning. The reason why we created this application, we offer both student and teacher to interact through the quiz application while this process like pandemic etc.

# Goals for the Project

# This application will be desktop based with several quiz features. It has two different user interfaces : Student Interface, Teacher Interface. In the beginning, you have to login to choose which interface that you will use. If you login as a student, you can solve the existing tests and then you have to submit it to your teacher, also you can reach the results which you solved before. While you are solving your test, you can find out the question’s true answer if you solved it wrong. If you login as a teacher, you will have so many different options such as create a test, merge two test, create a classroom and adding students to the classroom, analyze the exam results, import question by using text writing.

# Rest of the opportunaties:

* Dynamic question number for each test
* Different level for each quizzes
* Assignning different quizzes to each classroom
* Reliable teacher profiles by using serial key for each teacher

# 

# Motivation for the Project

We felt that this application was our obligation to help schools and universities during the pandemic times. Our team has an immense amount of knowledge when it comes to problem solving, programming, teamwork, communication and working as a group in a small time. Not only would we strive to give the quiz application what we desired and what they needed, but we will continue to make sure everything perfect for the application at its very best for the users. Each one of us will always and will continue to give our 100% and more to making everything exactly fluently and smoothly working without any error and bug.

Our team has amazing co-operation between each other. Every each of us has different coding and problem-solving skills. Also, having multiple heads working on the programming abilities helps in ensuring no errors are implemented and every single detail is put into application. We did not have knowledge about database, but we will work so hard to create a database fort his application which working without any delay and error. We encourage each other to not miss any step while we develop, and we help each other when one of us feels badly or fill every space of our fault.

# Process Flow Preview

For our process flow, we planned by following the software development stages. All requirements were discussed firstly, and a report of requirements and details of the project were formed. After the requirements are decided, design session started. In this session, use cases are tested and the UML diagrams were created. A report of the design session was formed again to show design to instructor. We wish to make sure that if we miss anything, we do not figure that out in the construction stage and changes of the requirements, or the additional improvements were added to design again. After planning requirements and design session software implemented with the changes and software were tested while coding and the after the code session were completed.

**Analysis and Design**

*2.*

# Plan for Requirements Engineering

## Inception Task:

The goal for the beginning is to identify the educational case created by the lecturers. They introduced the project sample and requirements. Completely object-oriented based code design and dynamic GUI were in demand. In the beginning of assignment ,the project had to be on desktop environment. In the upcoming days, other platforms started to be available with project requirements updates. For desktop part, Java Swing was introduced by lecturers to be in use while project processing. Instead of Swing, our project team had already started to construct desktop application with JavaFX GUI library. Also, to get a basic understanding of the project, here were some more questions we asked:

*Which programming language must be used ?*

*What sort of object-oriented requirements they expect from project ? (abstract class, interface etc.)*

*Will there be any issues or constraints that may affect the planning and construction?*

*Is there any expectation for database management tools will be used?*

*Is there anything else you want to add?*

## Elicitation Task:

The goal at this stage is to identify the problem, propose solutions, and talk amongst each other on the many different approaches. Team is in the opinion of combining assignment with a customer request. Meetings are scheduled with customers team in order to get a more refined understanding. The plan is to get a grounded idea of what the objectives for the system are, what should be accomplished, and how the overall system fits into distance education .

## Elaboration Task:

In this stage, available frameworks and libraries were investigated and determined to use. Online lessons and resources were founded and took advantage of them. Several planning, scheduling, project management and code sharing tools were used. Also, online communication software was very helpful for teamwork during COVID-19 pandemic. Additionally, worked on simple and effective user interface model and features were determined. Main object classes and interfaces were discussed to fit in the project appropriately.

## Negotiation Task:

With any conflict that would come up in need of a resolution, the team would discuss them through to come up with a resolution. If there are too many requirements asked by the lecturers, team would have them rank each by their importance. Also, customer demands were ranked by team to complete project in time and some requirements were added to additional features list.

## Specification Task:

During this task, we plan to create a software requirements specification template. In this template we will note down the overall purpose of the project and the intended audience. Descriptions regarding the product features, user classes, operating environment, design and database to store data will be included. UML diagrams were created as requirements and requirements were coded as planned by report and diagrams. While coding the software changes of the requirements were updated and these updates added reports and diagrams.

## Validation Task:

During this stage, any requirements stated are to be ensuring that they are clearly defined. No miss-interpretation should be present and any that exist should be resolved. The application was tested during the software process through the eyes of the user. The necessary changes were made and approved. All exceptions were examined and misuse exceptions like input errors were checked and alerts of these errors were displayed in the application. All requirements and exception controls were validated.

## Requirements Management:

Any changes that may occur throughout the project stages should be handled with clarity and care. All change ideas of requirements were discussed and determined if the time allotted for the construction of the project can allow for such a change - that is if it is agreed upon by the project team. Requirements Management will occur throughout the project process flow as changes or alterations can occur under any circumstances. Due to project is presented as a term homework some changes was not able to apply. Additional changes assigned as future jobs

# Functional Requirements

## Hardware Requirements:

The software should be ran on any sort of desktop or laptop environment, necessity of the Windows operating system. The software has not the potential of running on tablets and mobiles. Essential input/output devices are keyboards, mouse, and printers; nothing else is required but can be recommended if desired.

## Application Interface - Primary Tasks:

* + - View all actions that teacher and student can do
      * Shows all actions with different and simple main menu designs.
    - Create new quiz to assign
      * Input screen for quiz features and after quiz information shows question adding screen with answers and explanation.
    - Create classroom
      * Screen for teachers to create new classroom and add students.
    - Show old quizzes and scores
      * A screen will show old quizzes for each user (teacher/student). Another screen to show grades.
    - Solve new quizzes
      * A screen will show mew quizzes for student. Another screen to solve quiz question by question.
    - Allow for registration
      * Registration display form
    - Allow the teacher/student to log in
      * Log in display form

## Application Interface - Secondary Tasks:

* + - Allow for student deleting from classroom by teacher
      * Only for teachers who are registered, there were be a delete button on the classroom display page.
    - Allow for student adding into classroom by teacher
      * Only for teachers who are registered, there were be adding button on the classroom display page.
    - Authenticate any user logging in
      * communicates with the database to verify the inputted username and password is correct
    - Validation for emails
      * communicates with verifiers to confirm the input information on the email field.
    - Calculate scores through answers of students
      * algorithms implemented within the code that will calculate student’s scores
    - Deleting quizzes and merging two quiz
      * Option for teachers to manage quizzes with deleting and merging them. (Optional quiz name selection)
    - Alerts for unavailable actions
      * Alerts screen to warn user
    - Store user information in the database
      * When registering, the user will provide their information in the form - this information will be sent securely sent and stored in the database

## Assignment-side Software - Tasks:

* + - Application was constructed entirely object-oriented based
      * Abstract classes were used
      * Interfaces were implemented
    - Queue or stack data structure
      * Dynamic Queue was used in several classes
    - Responsive graphical user interface (GUI)
      * Desktop application was developed with JavaFX

# Non-Functional Requirements

## Performance Requirements:

* + - Optimization is convenient enough for every desktop environment
    - Speedy performance / transmission of data
    - Have a quick recovery time if anything were to go wrong
    - Display accurately and efficiently on every page

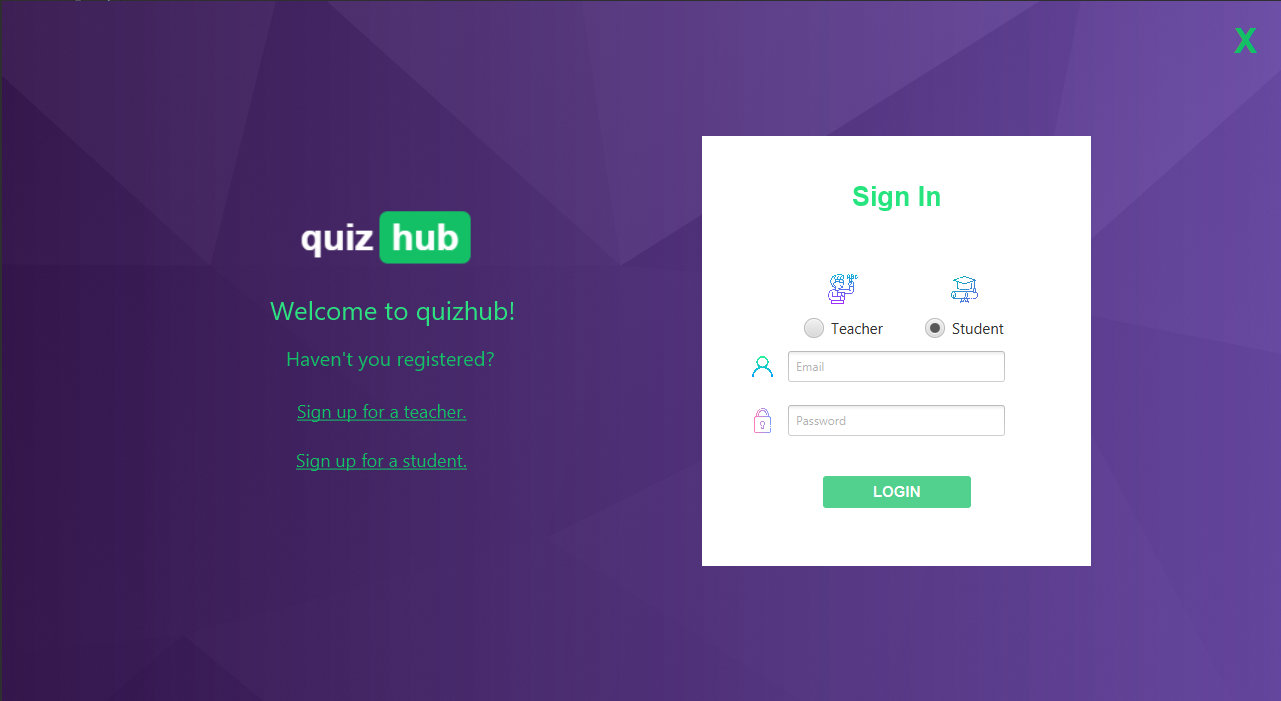
## Security Requirements:

* + - Unique serial key for each teacher product (only one use)
    - Prevent any potential threats such as SQL injections through the forms or search boxes.
    - Prevent third party users at administration level
    - Prevent false email inputs from being used when registering

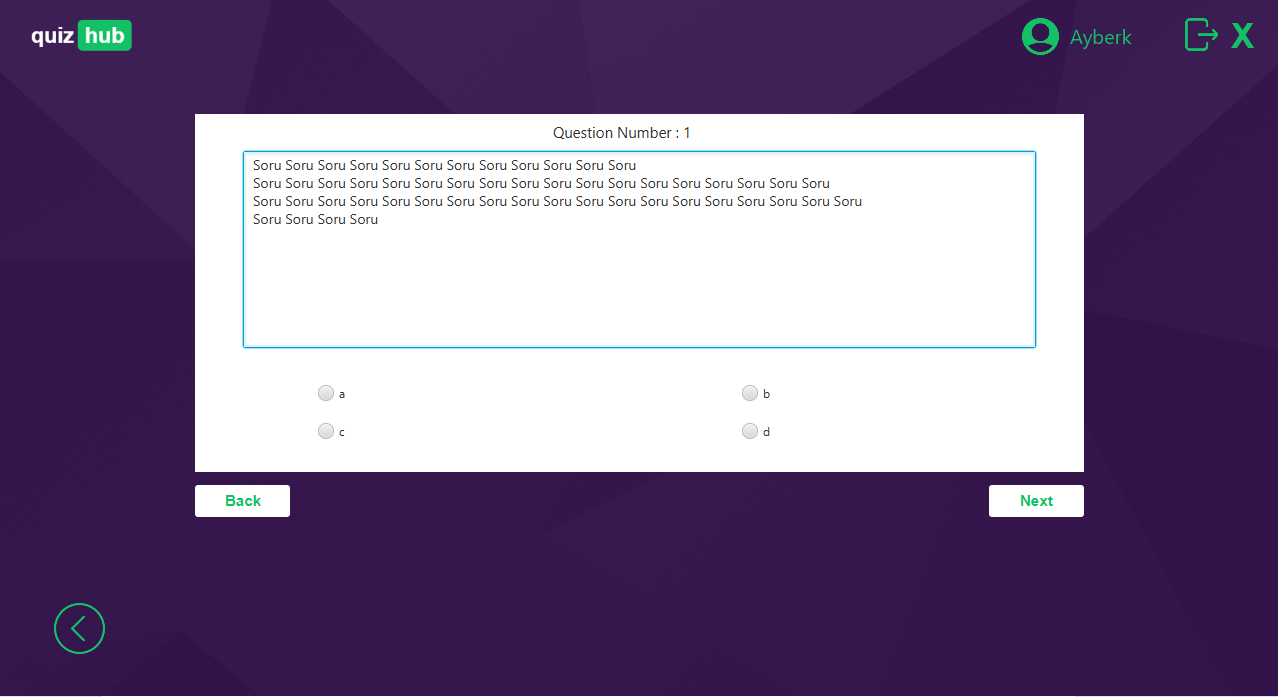
## Quality Attributes:

* + - Maintain a user-friendly environment that is visually appealing
    - Easy to see and use navigation
    - Maintain readable content
    - Different teachers’ information is not mixing with each other.

**Screenshot Mockups:**







# Use Cases

**Use Case #1: *Creating new quiz***

**Primary Actor:** Teacher

**Goal in Context:** Creating new quiz and assigning to classrooms

**Preconditions:** Quiz must be non-exist

## Scenario:

1. Teacher: Logs onto Quiz Hub application (Enters Email/ Password)
2. Teacher: Selects “Create Quiz” on application.
3. Teacher: Enters “Quiz Title”, ”Quiz Level” on new page.
4. Teacher: Enters question, answers, and explanation.
5. Teacher: Selects “Add” or “Finish”.

## Exceptions:

1. Quiz Hub Username/Password incorrect: Teacher is sent to login page to re-enter credentials
2. “Quiz Title” is not unique: Teacher is warned and sent to current page to re-enter
3. “Quiz Level” is not in range (0,5): Teacher is warned and sent to current page to re-enter
4. “Quit” is selected before “Finish”: Quiz is not created, Teacher is sent to main menu

**Priority:** Essential, must be implemented to assign quiz

**When available:** First Sign-up

**Frequency of use:** Optional for teacher **Channel to actor:** Quiz Hub Desktop App.

## Open Issues:

1. Should there be an option to time limit for quizzes?

**Use Case #2: *Solving new quiz***

**Primary Actor:** Student

**Goal in Context:** Solving quiz and getting a grade

**Preconditions:** Teacher has assigned the quiz to the classroom

## Scenario:

1. Student: Logs onto Quiz Hub application (Enters Email/ Password)
2. Student: Selects “New Quizzes” on application.
3. Student: Chooses a quiz on new page and select “Solve”.
4. Student: Answers the question and selects “Next” or ”Back” or “Finish”

## Exceptions:

1. Exit button is selected before “Finish”: Quiz is not saved, need to be solved again.

**When available:** Teacher has assigned the quiz to the classroom

**Frequency of use:** Optional for teacher

**Channel to actor:** Car Rental Website

## Open Issues:

1. Should there be an option to time limit for quizzes?

# Models

# A picture containing text, map Description automatically generated

**A picture containing text, map

Description automatically generated**

The programme has two different user interface: teacher user interface, student user interface. Each user types can apply particular operations.

Teacher usually interacts with the classroom, student and quizzes. Teacher has permission to add and remove student from classrooms. Also teacher can change directly quizzes such as creating new one, deleting...

On the other hand, student can not edit the quizzes and classrooms. Students can only reach the existing quizzes that created from their teacher and solve them in a particular time. In addition, they can see the past exams and scores that they solved before.

**A close up of text on a white background

Description automatically generated**

**Factory Pattern Explanation:**

*Program has two different user types(teacher,student). For user operations,* ***factory pattern*** *is used as mentioned in the class diagram above. Each user class extends “Person” abstract class that include general information such as name,surname, gender vs. Also, each classes has different attributes that related their user type. Simultaneously, these user classes implements the “User” interface. In this interface, there are two methods (insert(),delete()) to be overrode in “Student” or “Teacher” classes and one static “login()” method. With “UserFactory” class, the programme returns an user object with person type that student or teacher. To obtain a person (implements User interface) register page must call the getPerson() method in “UserFactory” class.*

Teacher class has some methods that interact directly database such as sign up, login. Student class also has sign up and login methods. Application has three different classes such as quiz, question, classroom and these classes have specific controller classes to manage user interface.

Classroom controller classes provide teacher to creates a new classroom, adds new quiz to selected classroom and displays quizzes and students of the selected classroom. Quiz controller classes provide display whole quizzes on the database, grading solved quizzes by the student and redirects to the add question page. Question controller classes provide display questions of the selected quiz and help the user to navigate in the questions of the selected quiz.

Register controller classes hold user’s informations and post these informations to the database and create a new user. Login controller classes provide user login process by entered inputs. Menu controller classes are connected to the all classes to provide transition between the pages and closing the page.

A screenshot of a cell phone

Description automatically generated

**Prototype Pattern Explanation:**

*In original project, database connections are not efficient because of usage in many different controller classes. For each quiz and question operations, program is used to connect database again and again. To make these processes more efficient, for quiz operations prototype pattern is used. The pattern is constructed with different quiz types; “****NewQuiz****” ,”****OldQuiz****”. NewQuiz is used for operations of creating new quiz. On the other hand, OldQuiz is used for operations of solved quizzes and already existed quizzes. Both of these different quiz types extend the* ***abstract*** *“Quiz” class that comprises common, identical attributes.*

*In the pattern, “****QuizCache****” is the most important class. It has hash table with the type of Quiz and from this hash table, program can construct database connection only once and store them in this hash table with id keys. Also, there are two different arraylists to store quizzes’ id that will be used as a key value in hash table. The reason of usage of these arraylists is quick access to quizzes corresponding to the of “solved” and “unsolved”. In this way, program does not need to traverse all hash table. Also, class has three different, significant methods. One of them is the “****getQuiz()****” which is the method to obtain any quiz from different classes with using hash table. The other two methods have crucial role for pattern, “****LoadCache()****” method is called after any login operation and first caching operation is completed. If teacher creates new quiz or delete a existed quiz or user logout, “****CleanCache****” will clean the hash table and arraylists. After clean operation, if user still logged in, LoadCache will cache quizzes again. In this way, there is no* ***id*** *confliction between database and main program.*

*In the program, several controllers can achieve the methods in QuizCache and operate them.*

*The foremost usage of pattern is displaying the quizzes on the user interface. So, as mentioned in prototype pattern class diagram above, number of controller classes use pattern to display quizzes with mentioned* ***methods*** *in these classes. Some of them uses LoadCache for lesser user waiting time, others use CleanCache to make hash table and arraylists ready to load again.*

**Teacher – Merge Quizzes Activity Diagram (NEW) :**

*A close up of a map

Description automatically generated*

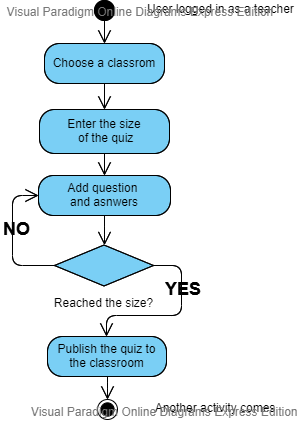
*In “Edit Quiz” section of application, teacher has option to merge quizzes that created before by same teacher. If teacher selects the merge quiz option, there will be a screen with two same quiz list. Teacher can select two quizzes from these quiz lists to merge. Also, if teacher does not give a name to new merged quiz, application will give a name by using two quizzes name that will be merged.*

**User – Registration State Diagram (NEW) :**

*A close up of a map

Description automatically generated*

*Login screen is the starting screen of application and user must login to use the application. If user has not a quizHub account, must create an account. There are two option to sign-up in login screen: teacher, student. After user select the account type, there are credential text field. If user fills fields wrongly or leave empty, application will warn the user. This sequence will continue to user enter his/her credentials correctly. After registration, user is directed to login page again.*

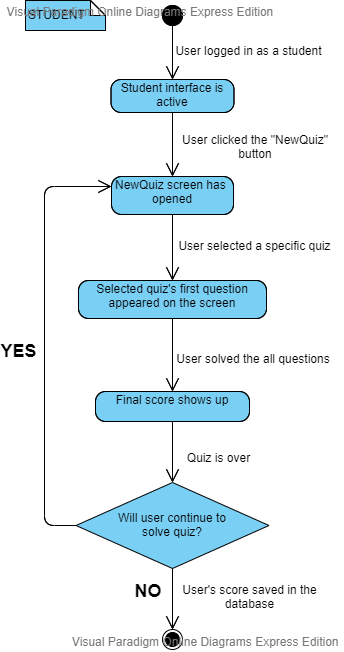
****

First of all, user has to login as a teacher to create a quiz. After validation of user type, teacher must decide the size of the quiz such as 20,30,40 and so on. After deciding size of the quiz, teacher must add questions until he/she reaches the maximum question limit. Finally, all adding statements done, teacher has to upload that quiz to the classroom then the students are able to see that quiz on their screen.

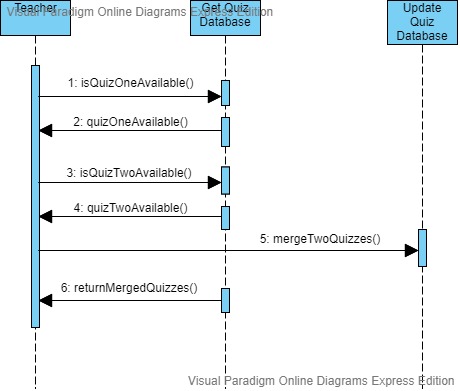
A screenshot of text

Description automatically generated

This diagram explains about adding question to the quiz by teacher. User has to import image as a question or write a question by using keyboard. After that, user has to decide 4 different answers, one of them has to be true and the rest of them must be false answers. Then user has to write an explanation for that answer because student needs to learn why his/her answer is wrong. Finally, user must decide each question's value in range 10 because every questions dont have to be same difficulty. Student should see the value of the question in order to difficulty of the question so on he/she can measure own knowledge.



This state diagram explains the processing of solving quizzes by a student user. Firstly, user who has been signed up as a student logs in and student user interface is opened by this log in. Then, the user chooses a quiz to solve. After solving all questions in the quiz user's quiz score shows up and quiz is over. Finally, user chooses whether continue to solve quizzes or exit application.

****

This sequence diagram explains combining of two or more quizzes by the teacher user. All quizzes in this application are kept in the database so program checks the database to check for quizzes. If there is no error in this check process teacher can click the merge button. After that, database is updated by new merged quizzes and new quiz is being available to solve.

**Project Plan**

*3.*

* 1. **Task Descriptions**

## Lecturers and Customers Meetings

The stakeholders and our team conduct meetings in order to get the full

grasp of the problem at hand, obtaining any and all information needed. All functionalities which

properly needed in this application would be decided clearly with the customers. Conflicts and

necessities would be decided during this time and any time throughout the project process

flow.

## Design Models and Mockups

Designing the models and mockups help to ensure clarity in view of the project as well as how it

works. Lecturers gave us the functionality and necessities but all design and stuff like that was our

choice to create.

## Database Creation

A database is created using the models to provide storage for student information, teacher

information, quiz properties, information of the question, relationship between classroom and

student, relationship between quiz and student, relationship between classroom and quiz. Testing of

the database is ensured at this point.

## Employee Software Creation

The software that is to be used by the teachers and students will be designed using JavaFX

using the guide of the mockups, requirements, and theme templates for design. The software

will act as a simple to about creating and solving quiz functions and among other functions

the database – the information stored including student and teacher information, quiz

information, and classroom details with their people.

## Application Creation

The application will be designed using Java programming language using the guide of the

mockups, requirements, and models. The application will be implemented using JavaFX and Scene

builder for creating backend and frontend. It should allow for users to keep in touch through the

quizzes to grading like school or university. Teachers are able to create quizzes and assign this to the

specified classroom, students must solve these quizzes to get their grades.

## Testing

Testing will be implemented on software. Test cases may be used to realize are there any bug

or error for both user side. There were different bug and errors in the application but with

after our testing part these bug and errors resolved.

## Finalization and Reports

All testing and function processes are finalized at this stage. Reports will be created to ensure

all information and functionality is clear in order to make the user manual and to help ensure

employees can use the software with ease.

# Task Assignment

All three worked together in the project planning, sharing in the opportunity of any models

and analyzing all specifications made by the proprietor in the same time.

First of all, we talked with our stakeholders to learn application’s requirements and

necessities. After that meeting, every requirement and functionalities were clearly understandable.

We started with designing main user interface templates for teacher and student. While we are

designing these interfaces, our main classes created by us. After main interface designing part done,

we decided what should we do about teacher’s and student’s functions such as create quiz, solve

quiz, create classroom etc.. We created so many pages for every function to make application more

smoother and understandable. While we were creating the login and register page, we designed our

basic database which will be updated in the next weeks. We finished our main interface and page

designs by using JavaFX. We updated our database in order to our application’s requirements. We

started our testing to find our bug and errors. After testing statement, we found our bug and errors

and we fixed all of these.

Reports were created with all three members together after every milestone completed.

# Deliverables and Milestones

We had four major Milestones in this project:

* + 1. Completion of Requirements Gathering.
    2. Completion of Design and code.
    3. Testing database.
    4. Completion of Testing.

These milestones were all completed on schedule and yielded a Deliverable at the end of each. Our four corresponding deliverables (respectively) in this project were as follows:

1. Met with customers.
2. A finished and easily navigational GUI (Graphical user interface).
3. Responsive OOP based coding.
4. Satisfied lecturers and customers after demonstration and launch of software.

# Project Schedule

The first month of the project start date (1 March 2020) was used mainly for requirements

setting. This took almost a week or two weeks. When all the necessities and requirements

settled up, we started creating a interface with our design templates. After some page’s

designs finished, we started coding while we were designing also. We combined both coding

and designing interface because they had to be combined while creating. In April we started

to create main database for login and register processes. After nearly 4 weeks, our database

mainly created with some minor mistakes. In these 4 weeks, we kept coding backend and

finished main functions which needed by the stakeholders.

We finished our database and application connections without any error or bugs. On 25 May, our project completed without any error and report written by team.

The hours we spent for each state as follows :

Requirements: 15 hours

Design and code: 65 hours

Testing: 25 hours

Manual and Final Report: 10 hours

Total: 115 hours

**Testing**

*4.*

# Features to be tested

We will start by using both static and dynamic testing strategies. The static strategies will

include reviewing the basics of the application whereas the dynamic testing is based on actual

code execution.

The features we tested were as follows:

● To ensure that the application itself ran (Dynamic)

● Logins worked efficiently and consistently (Dynamic)

● Accessing the quizzes and questions from database (Dynamic)

● Creating quiz, classroom and assigning them to the database (Dynamic)

● Grading after solve quiz (Dynamic)

# Test Cases

The following are examples of test cases we implemented:

• Graphical User Interface

• All fields on page (e.g. text box, radio options, dropdown lists) should be aligned

properly

• Upon click of any input text field, mouse arrow pointer should get changed to cursor

# Testing Schedule

The testing should begin right after the project itself begins. Keeping up on testing will ensure

that any mistakes are caught early and corrected immediately while we are developing the application.

**Conclusion**

*5.*

* 1. **The Problem and Solution**

The problem that undergoes for online education applications was a lack of teacher-oriented practice. A teacher-based application was requested and

Another problem was the need for a quiz-based application that would facilitate distance learning in difficult times like pandemics to provide contact by question and answer methods. At this stage, the quiz system required to test the efficiency from distance education was applied. The teacher user should be able to easily create quizzes wherever they are, to assign these quizzes to created classes he/she chooses from his own students, and to view and edit these classes and quizzes as he wishes.

All class and quiz information should be kept in the database and should be open to access whenever the user wants to access it. When this data is edited, it should also be changed in the database.

The solution was to provide quiz application with a user-friendly desktop application that would allow for teacher users to create classrooms on the system by chosen students and create quizzes by typing the questions, displaying all questions of the chosen quizzes and editing these quizzes by delete or merge process, assigning quizzes to the chosen classroom which is created by same teacher and managing the classrooms by deleting and adding students and for the student users to solve assigned quizzes to her/him classes and displaying grades of the solved quizzes and the true answer explanation of the questions of solved quizzes. The application stores all data such as questions of the quizzes, students of the classrooms and the created classrooms and quizzes of the teachers. This software provides an easy-to-use interface to allow for simple access to quiz, classroom process

* 1. **The Team and the SE Process**

The Software Engineering process we used was the pair programming method. Pair programming is an agile software development technique in which three programmers work together at one workstation. One, the driver, writes code while the others, the observer or navigator, reviews each line of code as it is typed in. The three programmers switch roles frequently. In this method, we started the discussion requirements and detailed design together and we coded the software together in ordered cycle and end of the development we tested the application together. We used necessary utilities like discord to use this method in the pandemic process. A coder shared screen and the others help him to search necessaries and solving bugs and coder coded the software by discussion. The coder changed in ordered cycle and all coded software is uploaded to GitHub. Knowledge is constantly shared between pair programmers. Pair programming allows team members to share problems and solutions quickly. This helps pair programmers to learn to communicate more easily.

* 1. **Engagement of Umbrella Activities**

Four of the main Umbrella activities we used were as follows:

* + 1. Software Project Tracking and Control – is used to lead the project and ensure that the project was controlled and on schedule.
    2. Reusability Management - was used to help us create flexible and generic assets that may be reused for future projects or for this project in other regions. Object oriented based programming has the main role in this activity.
    3. Software Configuration Management – is provided to control of effect changes of future works of the project on schedule.
    4. Risk Management - This activity was used to assess and identify potential risks with creating the software such as assuring that not too much money be spent in assets on the project.
  1. **The Stakeholders that Benefited**

After release of the product, all of our active stakeholders benefited from the software. The users as teachers or as students benefited the software in distance learning at facilities like pandemics. The programmer also benefited software to experience of creation an application for future works.

* 1. **The Organization's Benefits**

Our Organization benefitted greatly from the production of this software. With this organization, we have experienced the stages of application development in more detail.

We had the experience of using interface and database for our future works. If the software and applications are liked, we will be more motivated for our future works. Also, the feelings of the creating an application may affect to choose orientation of our future jobs.

**User Manual**

*6.*

* 1. **Software Description**

The desktop application will allow for the teacher and students to keep in touch by creating and solving quizzes during the pandemic situations like corona etc. This application is useful for teacher-student grading for distance learning. There are 2 different user types like student and teacher.

The teacher can create a classroom and should add students to this classroom. After that, teacher can create quiz as much as he/she wants to assign to the classrooms. Teacher is able to reach the old quizzes which created before and remove the student from a specific classroom. Additionally, teacher is able to remove a quiz from classroom and merge two different quizzes which he/she selected.

The student cant assign himself/herself to the classroom, teacher must assign them to the classroom. After assigning statement student is able to solve the quizzes but student cannot create a quiz. If student finishes the solving, student’s grade will appear on the grades screen and every student can reach their past grades and old quizzes which they solved. Student also able to see classrooms which he/she belongs.

* 1. **How to Use the Software**

The desktop application designed for the teacher and students. All of the students and teachers must download the application to their personal computers. User must clicked the application icon on the desktop. There will be a login screen when software opened. If user is a newbie then user has to create an account to login. There are two different opportunity to register statement. One of them for teachers and the other one is for the students. After the registration part finished, user is able to login to the main application.

The software when user logged in will be split into four main sections. These sections different for teacher and student. Teacher’s sections are create quiz, classroom, old quizzes and edit quiz. Student’s sections are solve quiz, classroom, grades and old quizzes.

If teacher clicks the create quiz section, there will be a screen appear. On this screen teacher decides quiz’s name and level. After that, teacher starts entering question, answers and explanation of the question until teacher decides to finish creating part. When creating part done, teacher should assign this quiz to the classrooms which he/she wants. If teacher clicks the edit quiz section, there will be a screen with two different sections : merge quiz and delete quiz. Teacher can merge two different quizzes and delete a quiz.

If student clicks the solve a quiz section, there will be a screen appear. On this screen student must select the quiz then student is able to see the first question of the selected quiz. Student keep solving questions until the quiz solving done.

* 1. **Troubleshooting Common Problems**

## The Application

Problem: Invalid Login

* + - The credentials used to log in was not found in the database, input the credentials again in case of mistype

Problem: Invalid Register

* + - The credentials used to sign up was not valid inputs, input the credentials again in case of mistype

Problem: Invalid Selection

* + - The selection from list is unavailable, select again the element correctly and click the show / display / delete / add button.

Problem: Empty Input

* + - Check the input field, if it is empty fill it and try again otherwise there might be a database problem, try later.