

VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH UNIVERSITY OF TECHNOLOGY



Web Programming Project (CO3049)

Multiple-choice Test Management

Instructor: Mr. Nguyễn Đức Thái

Students

Name	Student ID	Class
Nguyễn Ngọc Hòa	2052485	CC02
Dương Ngọc Quang Huy	2052489	CC02
Trịnh Công Vũ	1852882	CC02

Ho Chi Minh city, 11/2022

Contents

1.Introduction	1
2. Requirement analysis	1
3. Use-case Diagram	2
4. Activity Diagram	4
5. Class Diagram	7
6. Design	8
7. Product development	12
8. Release and Maintenance	12

GROUP CONTRIBUTION

Name	ID	Role
Nguyễn Ngọc Hòa	2052485	Back-end
Dương Ngọc Quang Huy	2052489	Front-end
Trịnh Công Vũ	1852882	Back-end



1. Introduction

Exam preparation is always a difficult and time consuming job for teachers. In addition to having to create reasonable exam questions for students, they also have to manually align and format in Word. Understanding that, our team has devised a solution, a website to manage and create school-level exam questions, in order to support the teachers during the examination period.

In this project, we are going to build a website for making exam questions and exporting the questions in pdf format as an exam sheet.

2. Requirement analysis

2.1. Functional requirements

Requirement ID	Requirement Description
F01	Register
F01_01	Users can register with an available valid username and password
F01_02	The system can catch and display any error messages in registering an account.
F01_03	The system can save the account information (username, password, etc.) in the database.
F02	Login
F02_01	Users can log in to the system by the registered account
F02_02	The system can catch and display any error messages in logging in.
F03	Display resources
F03_01	The system can display the information of teachers
F03_02	The system can display the available courses

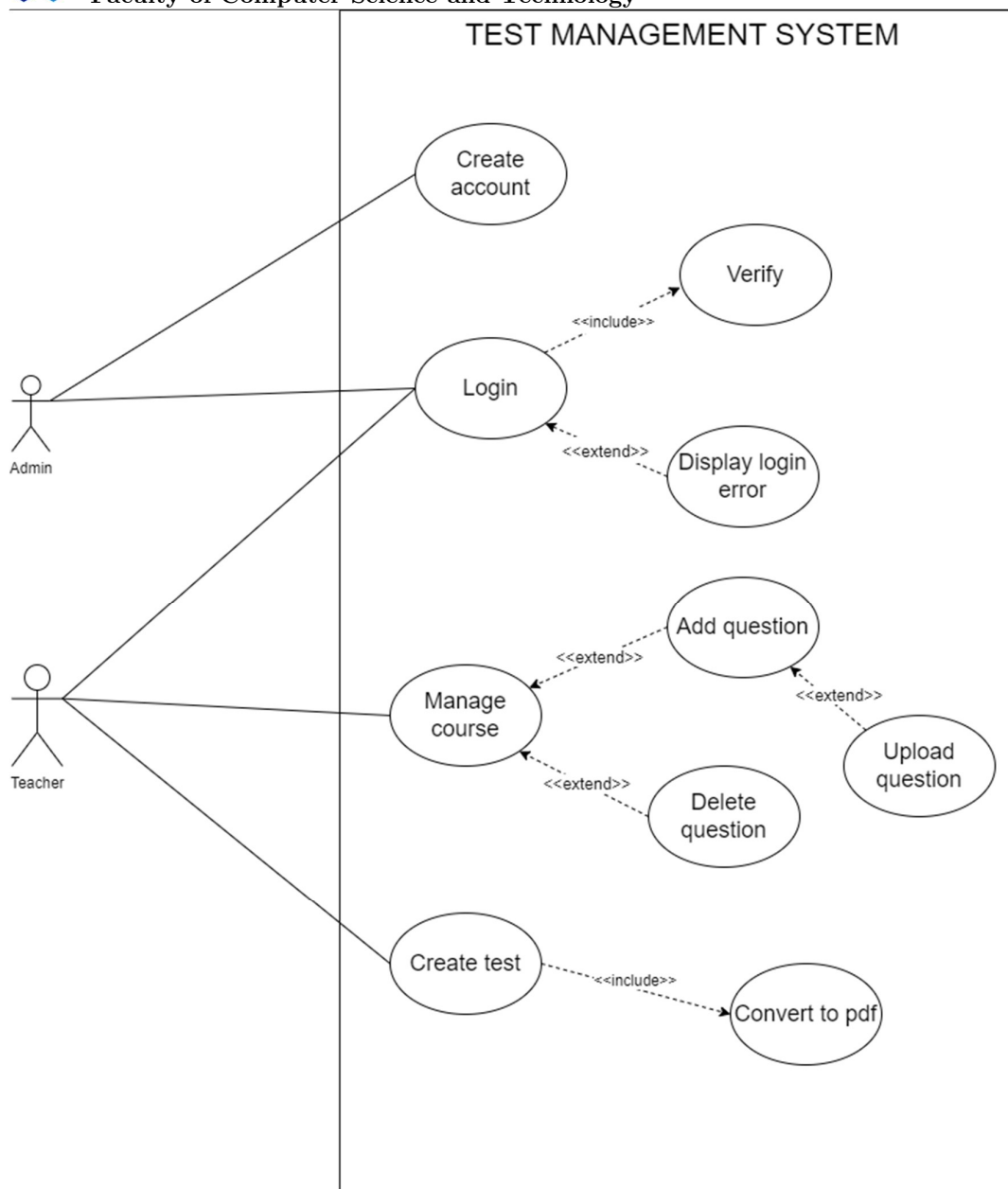


F03_03	The system can display the detailed questions in each course to generate tests.
F04	Logout
	Users can log out of the system.
F05	Test create
F05_01	The users can create questions for each course
F05_02	The user can collect and export the questions into pdf files as exam sheets.
F06	Update information
F06_01	Users can change/update personal information as well as password.
F06_02	The user can update and change the information of user, courses, and questions

2.2. Functional requirements

Requirement ID	Requirement Description
NF01	Operation
NF01_01	The system can be accessed when one has an internet connection.
NF01_02	The system can be accessed on any device (android, IOS, etc.).
NF01_03	The system has a friendly interface to use.
NF02	Performance
	The system can be accessed at any time (24h/day 365 days/per year)

3. Use-case Diagram



For the Online test management system, it will have 2 main features, which is “Manage course and its questions” and “Create new test from questions”.

Manage course and its questions:

- Create new course
- Add question to course (teacher can manually input question or use the upload question feature)
- Delete question from course



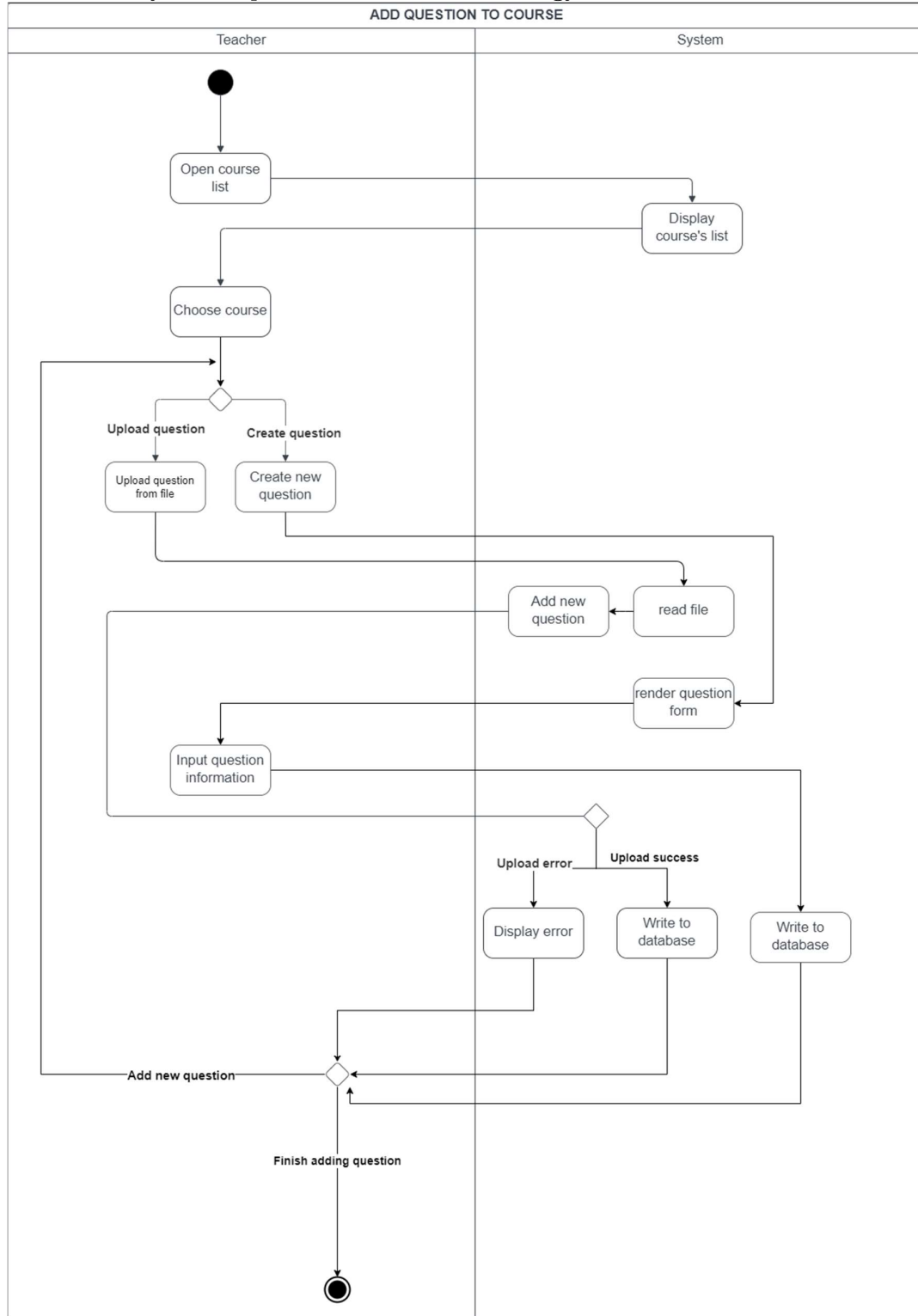
Create test:

- Add question to test
- Convert to pdf

Teachers will be provided a login account to use the Test Management System by the admin. Here, they can create their own course and test, which can be installed in pdf format.

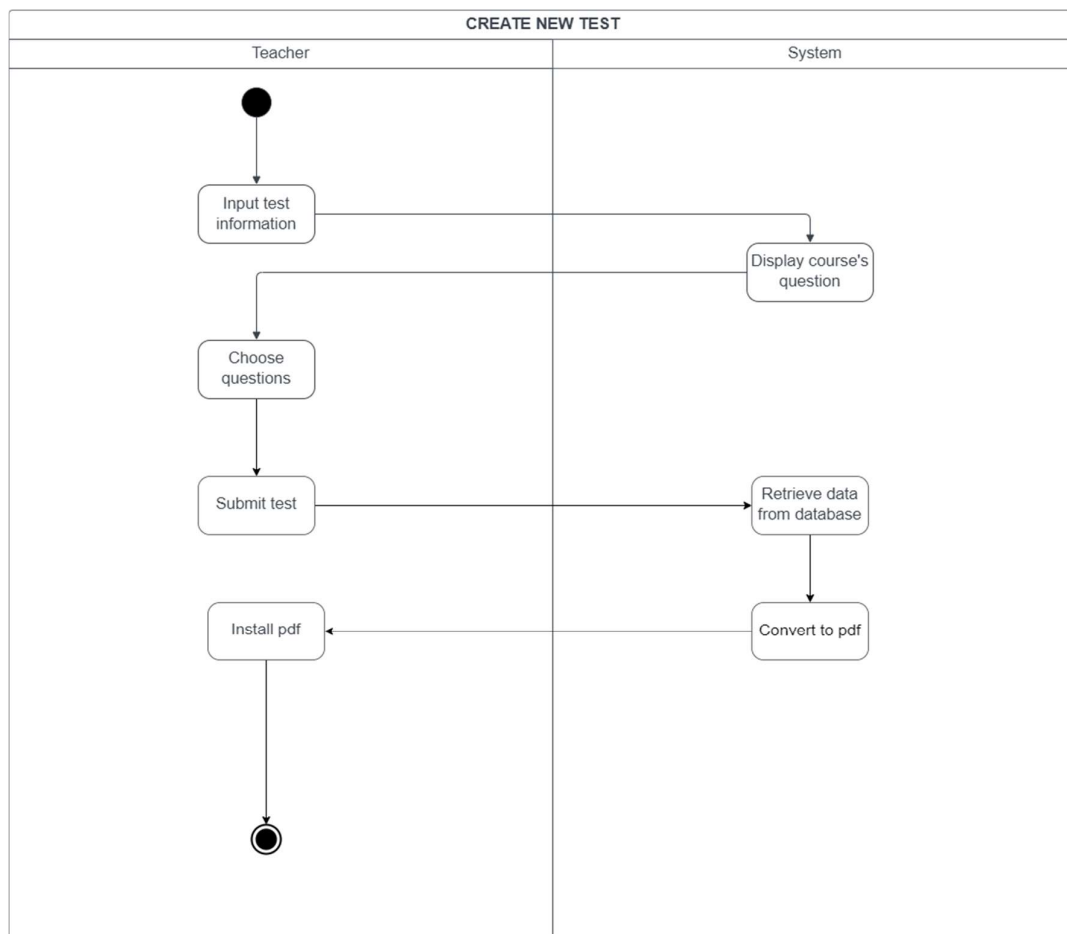
4. Activity Diagram

4.1. Add new questions to course



- Teacher will choose the course from the course's list that he/she created.
- In the course's page, the teacher can see all the questions of this course, they can add more questions by manually input or upload questions from excel file.
- If they choose to manually input a question, they will input the question, options and the correct answer as well as their difficulty level.
- If they choose to upload from a file, they will upload the file with the given format. The system will check if the file is valid and insert new questions to the course.

4.2. Create new test

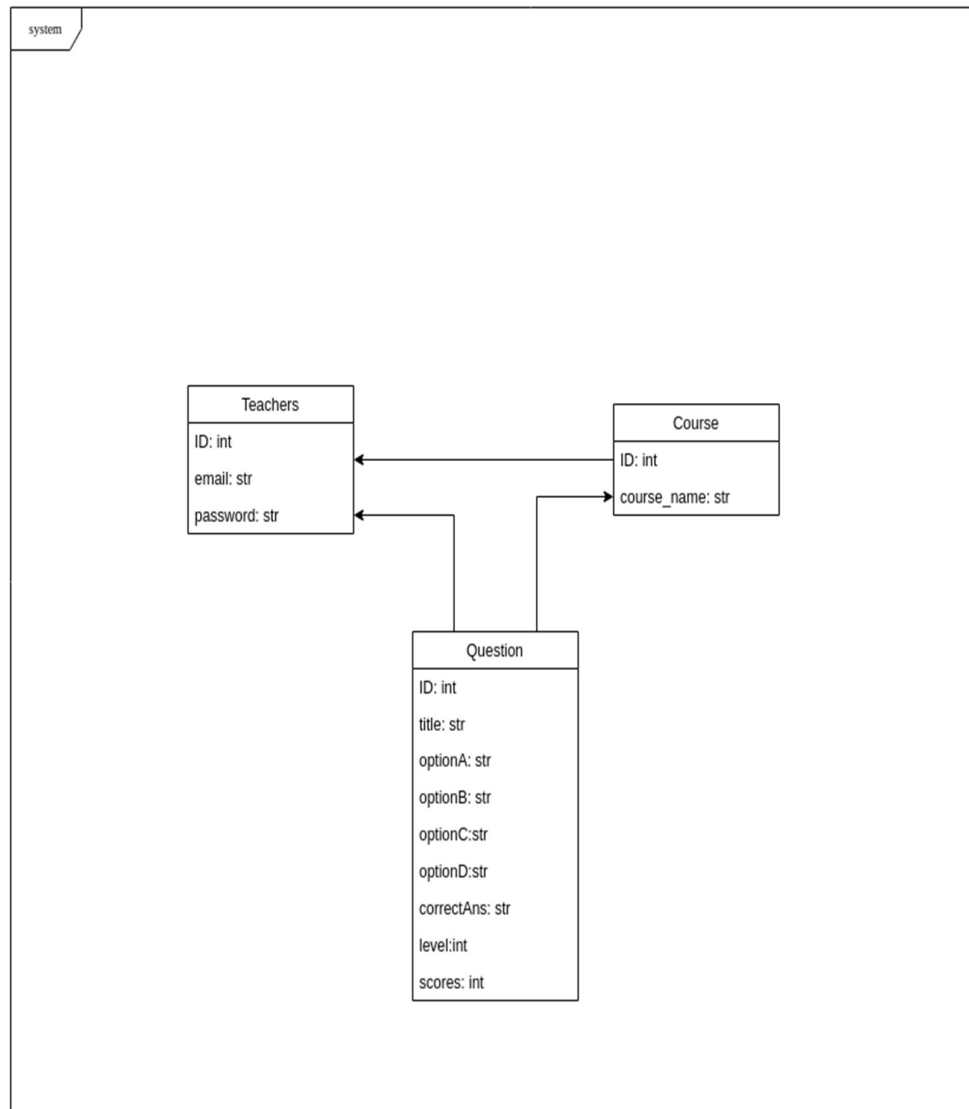


- To create new test, teachers will have to input test information (test name, course name, exam time)
- The system will provide teachers a list of questions from the course that they have chosen above.



- Teachers can begin to add or remove questions from the test, they can search and sort based on difficulty level.
- When done choosing, teachers click submit and a pdf file will be generated and ready to be installed.

5. Class Diagram



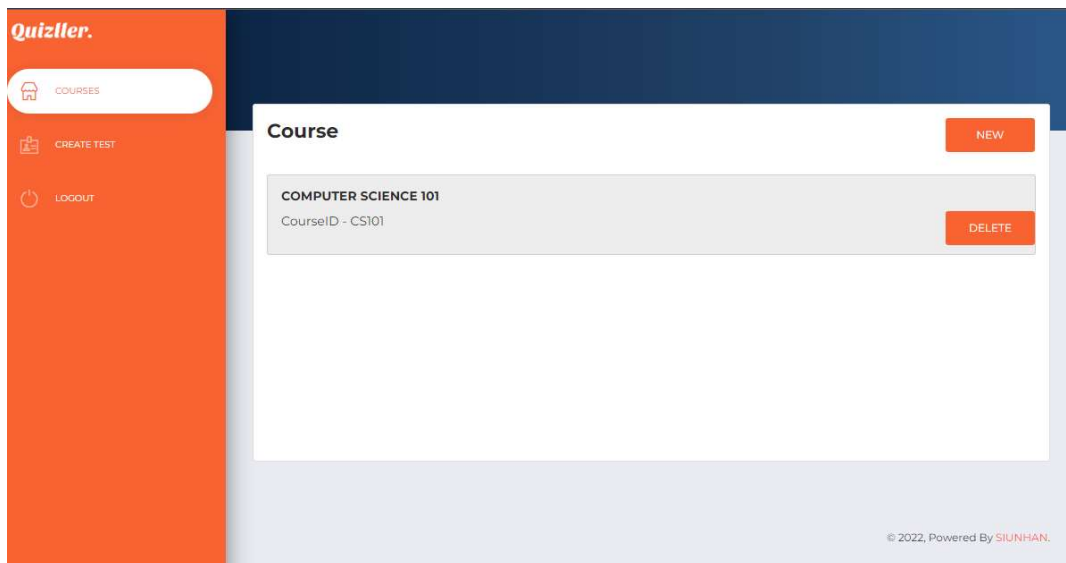
- The system consists of 3 models which are Teachers, Course and Question model. Each model represents a data table in the database.



- The **Teachers** model includes the **id** as primary key, **username** and **password** as the authentication keys.
- The **Course** model includes the **id** as primary key and **course_name** to distinguish between different courses.
- The **Question** models includes the **id** as primary key, **title** for the question description and 4 options **optionA**, **optionB**, **optionC**, **optionD** as the choices in the question, the **correctAns** field keep the correct answer for the question, the **level** field denotes the difficulty level of the question.

6. Design

6.1 Interface design



This is the layout of our website, we have the left sidebar to display main feature of the website.

The color scheme we use is Orange-White-Dark Blue, the color Orange will be mainly used to illustrate the part of the website where user can interact with (button, sidebar) because in our color scheme, orange is the most standout color, it will attract user attention. The White will be displayed where the text, content is and the Dark Blue will just be put at the top of the website to draw attention because it together with White create a perfect contrast couple.



Quizler.
 COURSES
 CREATE TEST
 LOGOUT

COURSE DETAILS

Course Name

Course ID

UPDATE

Test Questions

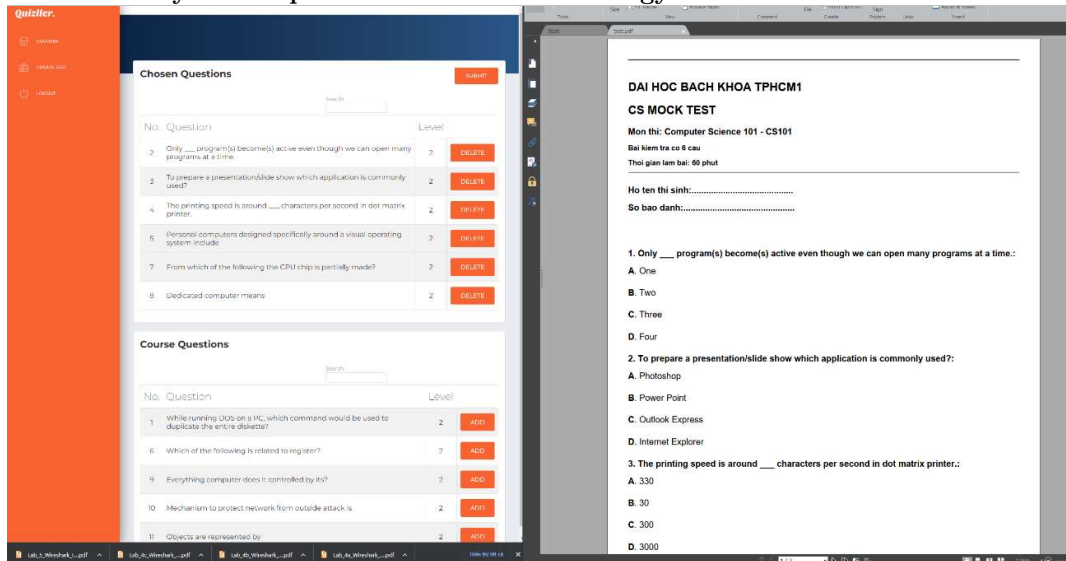
UPLOAD ADD NEW QUESTION

Show entries Search:

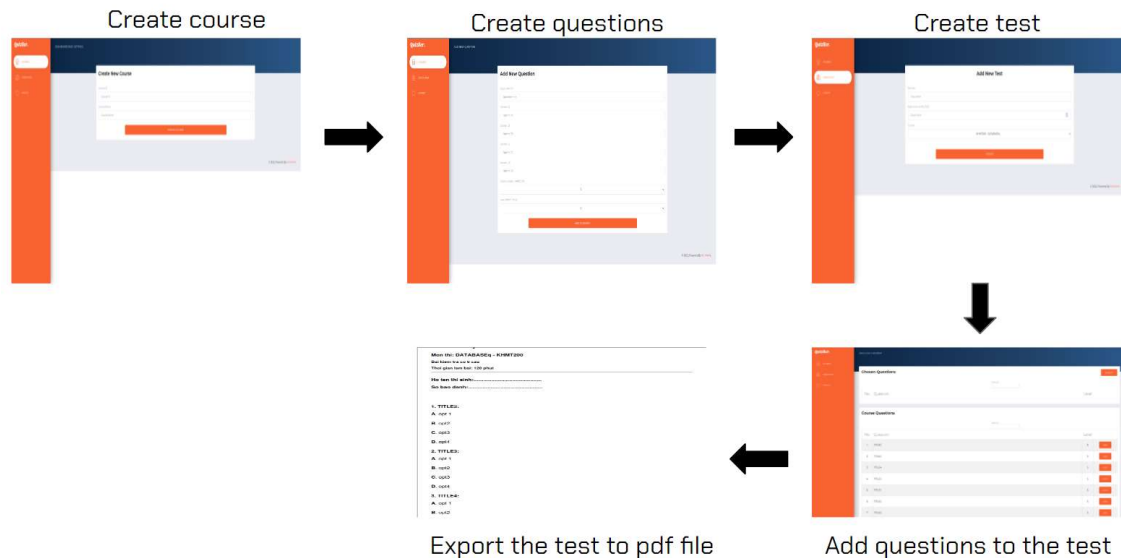
No. ^{↑ ↓}	Question	Level ^{↑ ↓}	
1	While running DOS on a PC, which command would be used to duplicate the entire diskette?	2	DELETE
2	Only ___ program(s) become(s) active even though we can open many programs at a time.	2	DELETE
3	To prepare a presentation/slide show which application is commonly used?	2	DELETE
4	The printing speed is around ___ characters per second in dot matrix printer.	2	DELETE
5	Personal computers designed specifically around a visual operating system include	2	DELETE
6	Which of the following is related to register?	2	DELETE
7	From which of the following the CPU chip is partially made?	2	DELETE
8	Dedicated computer means	2	DELETE
9	Everything computer does it controlled by its?	2	DELETE
10	Mechanism to protect network from outside attack is	2	DELETE

Previous 1 2 Next

The example pdf file will be:

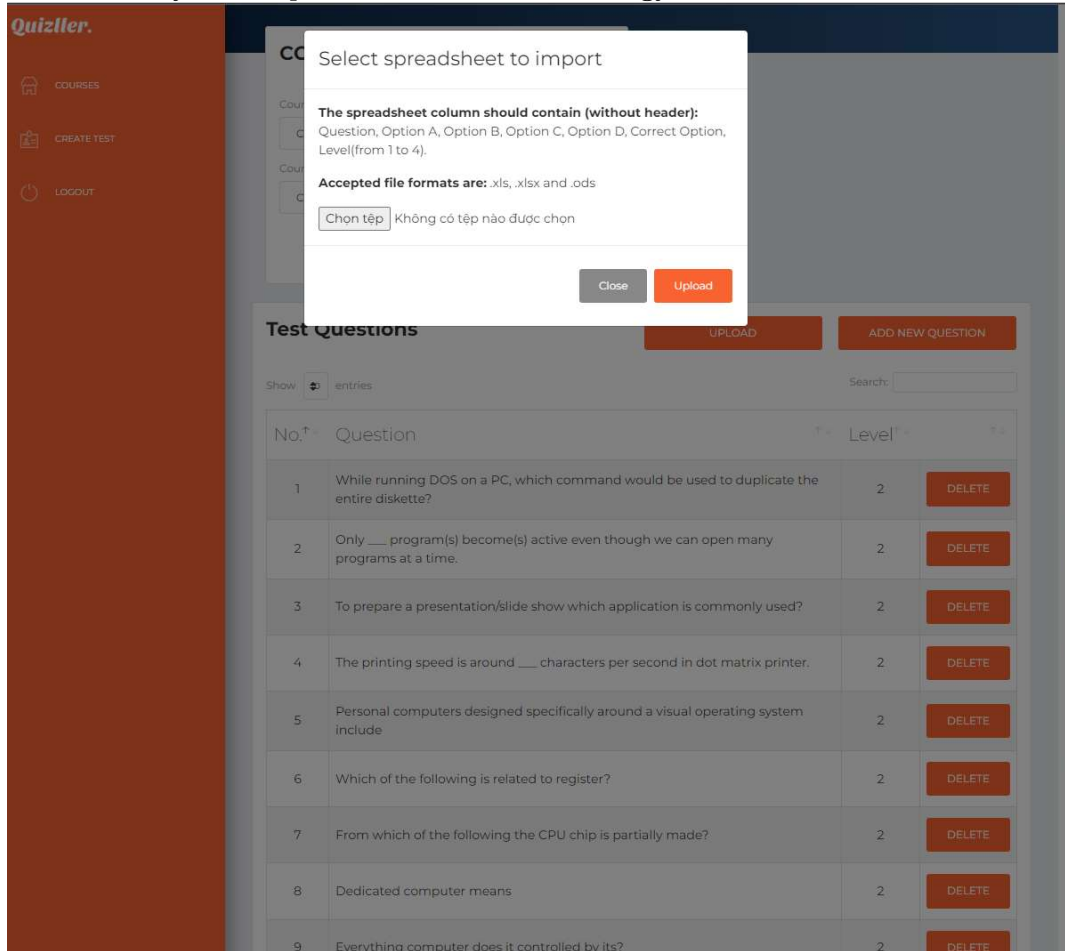


6.2 Interaction design

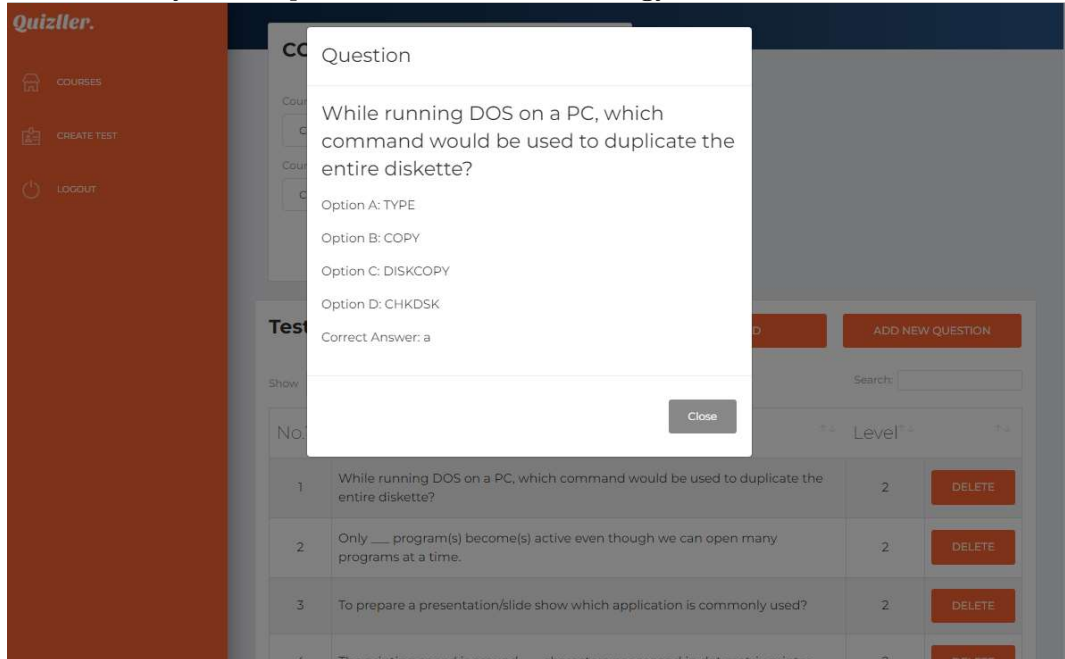


We try to minimize the step that take a user to create a test (which is the main feature of the website). With Quizler, user can create a test.pdf with only 5 steps through our pages.

For the Add question feature, we add the Upload option so user can easily add multiple questions at the same time with one click



To view the question, user can simply click on the question, a modal will appear with all the content of the selected question.



7. Product development

We decide to use PHP and Bootstrap to implement Quizler website and Mysql as the database.

The project source code can be viewed on this repository:

<https://github.com/siunhanvjp/BTL-WEB.git>

8. Release and Maintenance

8.1. Applying SEO

a. Title tags

In our website, for every pages that we show to the customer/audience, each of them all has a short and meaningful title that can fully describe what this page is about and contain the keyword of our website “Quizler”.

Teacher Dashboard:



```
<head>
<meta charset="utf-8" />
<link rel="apple-touch-icon" sizes="76x76" href="../../assets/img/apple-icon.png">
<link rel="icon" type="image/png" href="../../assets/img/favicon.png">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
<meta name="robots" content="noindex">
<meta http-equiv="pragma" content="no-cache" />
<meta http-equiv="expires" content="-1" />
<title>
| Quizller Dashboard
</title>
<meta property="og:site_name" content="Quizller"/>
<meta property="og:title" content="Teacher Dashboard"/>
<meta property="og:description" content="Dashboard for teacher view the current course, create, delete or edit course and question"/>
<meta property="og:type" content="website"/>
<meta property="og:locale" content="en"/>
```

Teacher's Course:

```
<head>
<meta charset="utf-8" />
<link rel="apple-touch-icon" sizes="76x76" href="../../assets/img/apple-icon.png">
<link rel="icon" type="image/png" href="../../assets/img/favicon.png">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
<meta name="robots" content="noindex">
<meta http-equiv="pragma" content="no-cache" />
<meta http-equiv="expires" content="-1" />
<title>
| Quizller-Course Details
</title>
<meta property="og:site_name" content="Quizller"/>
<meta property="og:title" content="Course Details"/>
<meta property="og:description" content="Show the details of course and manage questions here"/>
<meta property="og:type" content="website"/>
```

Teacher's Test:

```
<head>
<meta charset="utf-8" />
<link rel="apple-touch-icon" sizes="76x76" href="../../assets/img/apple-icon.png">
<link rel="icon" type="image/png" href="../../assets/img/favicon.png">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
<meta name="robots" content="noindex">
<meta http-equiv="pragma" content="no-cache" />
<meta http-equiv="expires" content="-1" />
<title>
| Quizller-Add Test
</title>
<meta property="og:site_name" content="Quizller"/>
<meta property="og:title" content="Add Test"/>
<meta property="og:description" content="Teacher create test with multiple questions for a course and install pdf file"/>
<meta property="og:type" content="website"/>
```

b. Meta tags

For each page, we also add a couple meta tags to briefly describe some information about that page such as content, description by using some Open Graph meta tags:

- og:locale - to define the language that we use for the website
- og:site_name - to define the main website's name, in our case it is Quizller
- og:description - description of that page.
- og:type - type of that page

- og: title - the title of that page

```
<head>
  <meta charset="utf-8" />
  <link rel="apple-touch-icon" sizes="76x76" href="../../assets/img/apple-icon.png">
  <link rel="icon" type="image/png" href="../../assets/img/favicon.png">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
  <meta name="robots" content="noindex">
  <meta http-equiv="pragma" content="no-cache" />
  <meta http-equiv="expires" content="-1" />
  <title>
    Quizller Dashboard
  </title>
  <meta property="og:site_name" content="Quizller"/>
  <meta property="og:title" content="Teacher Dashboard"/>
  <meta property="og:description" content="Dashboard for teacher view the current course, create, delete or edit course and question"/>
  <meta property="og:type" content="website"/>
  <meta property="og:locale" content="en"/>
```

c. Schema Markup

For our website, we use semantics elements in HTML to structure the content of the website:

- <nav> - defines navigation links
- <main> - defines the dominant content of the <body> of the website
- <aside> - defines the content in the sidebar
- <footer> - defines a footer of the website

8.2. Security

Considering the main feature of Quizller is to create and store teacher's courses and questions, our website will receive input from the user to meet their requirements, so our biggest security threat is to handle user input. Therefore, we perform "User input validation and sanitization" to increase website security.

The main point of validating and sanitizing user input is to protect the information in our database from SQL Injection. Therefore, when receiving the input, we strip tags, trim white space.



```
$title = strip_tags( trim( $_POST['title'] ) );  
$op_a = strip_tags( trim( $_POST['op_a'] ) );  
$op_b = strip_tags( trim( $_POST['op_b'] ) );  
$op_c = strip_tags( trim( $_POST['op_c'] ) );  
$op_d = strip_tags( trim( $_POST['op_d'] ) );  
$op_correct = $_POST['op_correct'];  
$level = $_POST['level'];  
$op_a = str_replace( array( '\\', "'", ',', ';', '<', '>' ), ' ', $op_a);  
$op_b = str_replace( array( '\\', "'", ',', ';', '<', '>' ), ' ', $op_b);  
$op_c = str_replace( array( '\\', "'", ',', ';', '<', '>' ), ' ', $op_c);  
$op_d = str_replace( array( '\\', "'", ',', ';', '<', '>' ), ' ', $op_d);  
$title = str_replace( array( '\\', "'", ',', ';', '<', '>' ), ' ', $title);
```

Before performing SQL query, we filter out the sensitive character that can harm our database by using a function called `mysql_real_escape_string()`

```
$title = "";  
if(isset($Row[0])) {  
    $title = mysqli_real_escape_string($conn,$Row[0]);  
}
```

Moreover, we also ensure that the user will input the right format to meet the website requirements by using regex

```
<div class="form-group">  
    <label>Password</label>  
    <input type="password" name="password" class="form-control" pattern = "(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}" title="Must contain  
    <span class="invalid-feedback"><?php echo $password_err; ?></span>  
</div>
```