

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

ĐH \* ĐHT



**ỨNG DỤNG TẠO TỰ ĐỘNG  
ĐỀ THI TRẮC NGHIỆM**

Sinh viên thực hiện:		
STT	Họ tên	MSSV
1	Trương Quang Đạt	21522922
2	Đinh Hoàng Thùy Linh	21522900

**TP. HỒ CHÍ MINH – 10/2023**

## 1. GIỚI THIỆU

- Đề tài này giúp tạo tự động đề thi trắc nghiệm, làm bài thi, và thống kê, xuất các kết quả kèm theo.
- Nhóm sử dụng các công cụ: JavaFX tạo mã nguồn, Scene Builder thiết kế giao diện, kết hợp thư viện JDBC kết nối cơ sở dữ liệu ,JDK, Jasper Report xuất báo cáo dạng PDF, SQL server lưu trữ dữ liệu.
- Kết quả đạt được: Phân quyền 2 kiểu người:
  - Giáo viên: Tạo, xóa, sửa đề thi; Thêm, xóa, sửa Ngân hàng câu hỏi, xem báo cáo, xuất đề thi dạng PDF.
  - Học sinh: Thi và xem kết quả bài thi, quyền sửa mật khẩu, nickname.

Cam kết:

- Đề tài do nhóm tự phân tích thiết kế và không dựa trên đề tài nào khác.

## 2. MÔ TẢ CƠ SỞ DỮ LIỆU

Cơ sở dữ liệu (CSDL) trong hệ thống Kiểm Tra trắc nghiệm của nhóm gồm 8 thực thể (Entity) sau: Người dùng, Học kỳ, Chương, Đề thi, Ngân hàng câu hỏi, Ngân hàng câu trả lời, Chi tiết đề thi, Kết quả thi.

Bảng Người dùng – TABLE tblUser

Tên thuộc tính	Kiểu dữ liệu	Mô tả
IdUser	char(6)	Khóa chính
userName	varchar(140) unique	Tên người dùng, dùng đăng nhập, không được trùng
realName	nvarchar(140)	Tên người
pass	varchar(20)	Mật khẩu
isAdmin	bit	Yes/No quyền người dùng

Bảng Học kỳ – TABLE semester

Tên thuộc tính	Kiểu dữ liệu	Mô tả
idsemester	char(4) ,	Khóa chính
namesemester	nvarchar(140),	Tên học kỳ

Bảng Chương – TABLE chapter

Tên thuộc tính	Kiểu dữ liệu	Mô tả
----------------	--------------	-------

idchapter	char(4) ,	Khóa chính
namechapter	nvarchar(240),	Tên chương
idsemester	char (4) not null,	Khóa ngoại Bảng học kỳ

Bảng Đề thi – TABLE exam

Tên thuộc tính	Kiểu dữ liệu	Mô tả
idexam	INT IDENTITY (1,1) ,	Khóa chính, tự động điền
idsubexam	nvarchar (140) unique	Mã đề thi không trùng
nameexam	nvarchar (140)	Tên đề thi
quantity	intquantity int	Số lượng câu hỏi trong đề
timeexam	inttimeexam int	Thời gian làm bài
datecreate	smalldatetime	Ngày tạo đề
idsemester	char (4)	Khóa ngoại Bảng học kỳ

Bảng Ngân hàng câu hỏi – TABLE questionbank

Tên thuộc tính	Kiểu dữ liệu	Mô tả
idquestion	char (6),	Khóa chính
contentquestions	nvarchar (1400),	Nội dung câu hỏi
images	varchar (200),	Link ảnh (nếu có)
idchapter	char (4),	Khóa ngoại bảng chương

Bảng Ngân hàng câu trả lời – TABLE questionoption

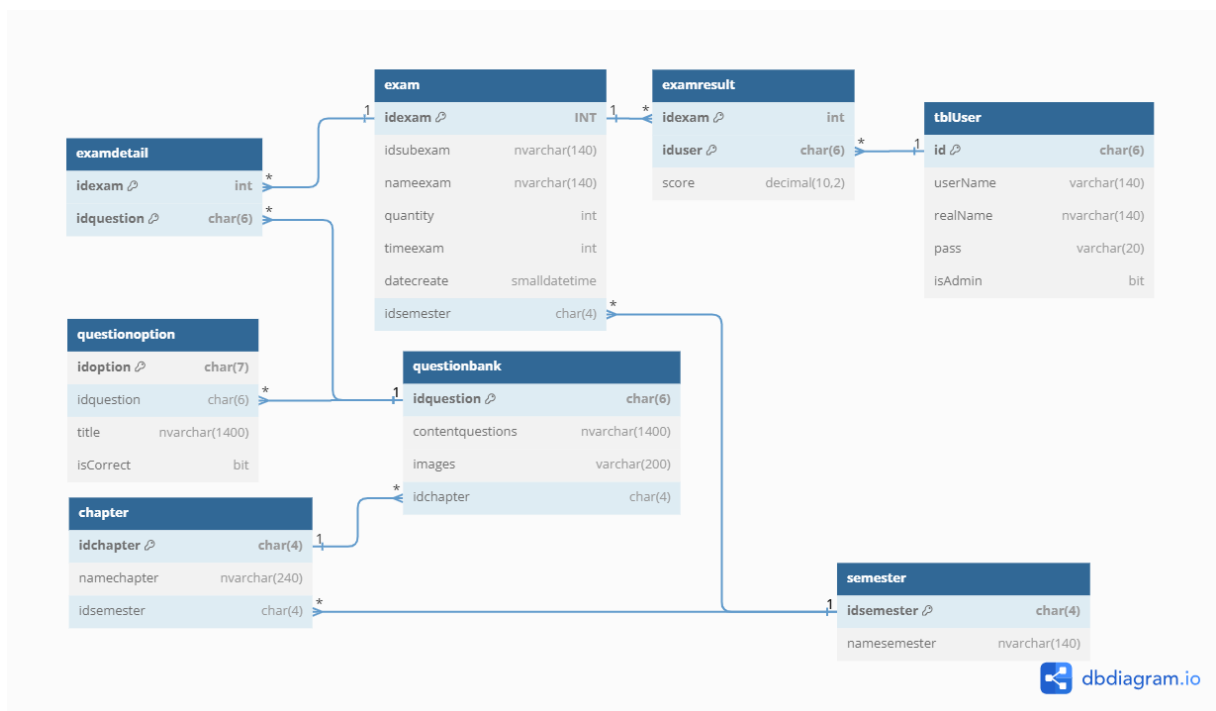
Tên thuộc tính	Kiểu dữ liệu	Mô tả
idoption	Int IDENTITY (1,1),	Khóa chính, tự động điền
idquestion	char (6),	Khóa ngoại bảng Ngân hàng câu hỏi
title	nvarchar (1400),	Nội dung câu trả lời
isCorrect	bit	Đáp án này là Đúng/Sai

Bảng Chi tiết đề thi – TABLE examdetail

Tên thuộc tính	Kiểu dữ liệu	Mô tả
idexam	int	Khóa ngoại bảng đề thi, tự xóa dòng khi đề thi bị xóa
idquestion	char(6)	Khóa ngoại bảng Ngân hàng câu hỏi

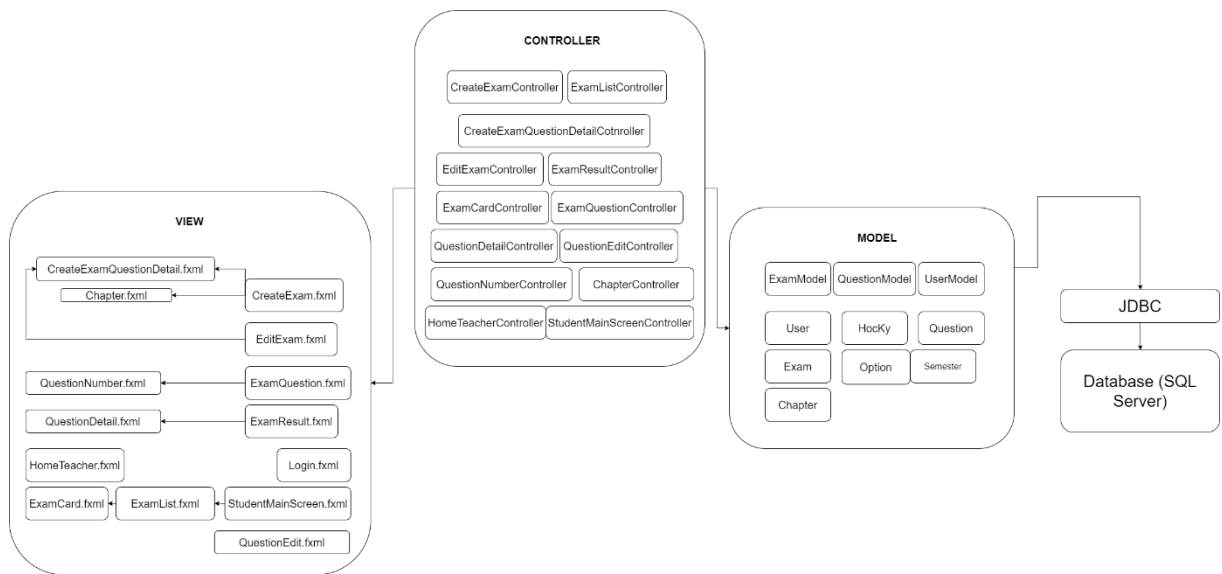
Bảng Kết quả thi – TABLE examresult

Tên thuộc tính	Kiểu dữ liệu	Mô tả
idexam	int	Khóa ngoại bảng đề thi
iduser	char(6)	Khóa ngoại bảng người dùng
score	decimal(10,2) DEFAULT 0.00	Điểm thi, tối đa 10 lấy 2 chữ số đơn vị, mặc định 0.00

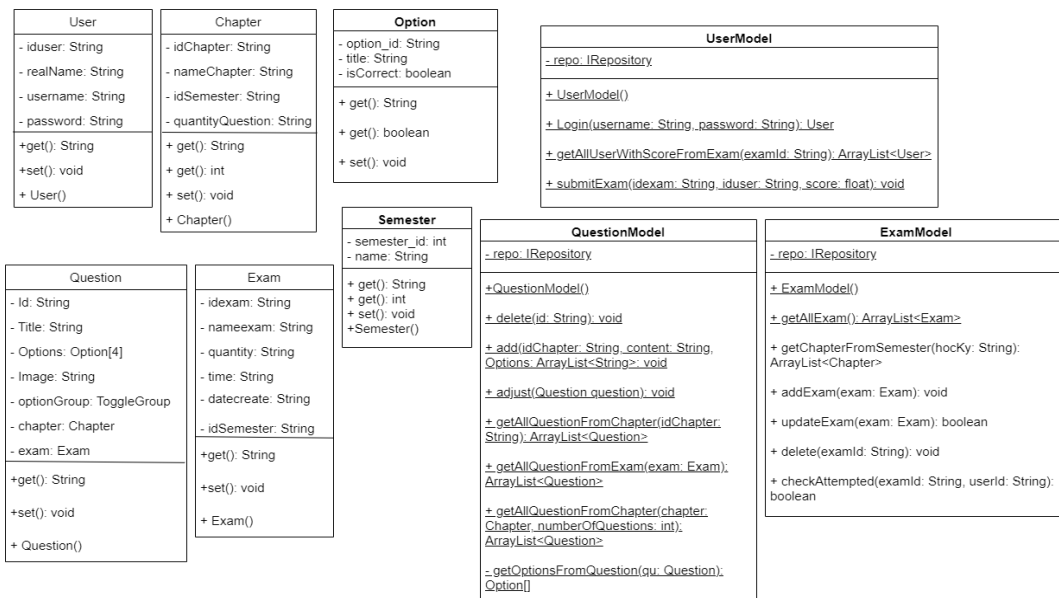


- Giải thích chi tiết:

- CSDL mô phỏng 1 năm học có 2 học kỳ, 1 học kỳ sẽ học nhiều Chương, trong mỗi Chương có nhiều câu hỏi có độ khó tương đương nhau, 1 câu hỏi sẽ có 4 lựa chọn nhưng chỉ 1 đáp án trong đó có thuộc tính isCorrect=True.
- Khi Người dùng có quyền Admin (isAdmin=true) đăng nhập sẽ có quyền tạo Đề thi. Chi tiết Đề thi được tạo tự động bằng cách dùng mã đề ghép với mã câu hỏi được lấy ngẫu nhiên từ ngân hàng câu hỏi.
- Khi Người dùng không có quyền Admin (isAdmin=False) đăng nhập sẽ thi theo các đề. Kết quả thi sẽ được tự động lưu xuống bảng kết quả thi.
- Kiến trúc hệ thống MVC: Nhóm thiết kế theo hình



## • Sơ đồ lớp:



### **3. THIẾT KẾ HỆ THỐNG**

#### **3.1. Mô tả ứng dụng**

##### **3.1.1. Công nghệ sử dụng:**

- JavaFX là một thư viện sử dụng trong ngôn ngữ lập trình Java, dùng để phát triển và phân phối các ứng dụng chạy trên máy tính để bàn và các ứng dụng Rich Internet Applications (RIA) chạy trên nhiều thiết bị khác nhau.

- Tại sao dùng JavaFX?

+ Vận hành đa nền tảng: Các ứng dụng sử dụng JavaFX có thể chạy trên đa dạng các nền tảng như: Windows, Linux, macOS và cả hệ điều hành Android, giúp cho việc phát triển app dễ dàng hơn và tiếp cận được nhiều đối tượng người dùng hơn.

+ Cung cấp nhiều tính năng để tạo giao diện đẹp mắt, thân thiện với người dùng.

+ Cộng đồng các người phát triển hoạt động tích cực, dễ học được nhiều kiến thức mới, giải quyết nhiều vấn đề.

- Hệ quản trị cơ sở dữ liệu MSSQL (SQL Server) là một trong những lựa chọn thích hợp để lưu trữ lượng lớn dữ liệu cần thiết cho hầu hết các sản phẩm công nghệ vì:

+ Hiệu năng và khả năng mở rộng cao - có thể xử lý các tập dữ liệu lớn và lưu lượng truy cập cao một cách hiệu quả.

+ Dễ sử dụng: MSSQL tương đối dễ sử dụng và quản lý. Nó bao gồm nhiều công cụ và trình hướng dẫn để giúp người dùng bắt đầu một cách nhanh chóng và dễ dàng.

- Đối tượng sử dụng:

+ User (Học sinh): Phần mềm Ứng dụng trắc nghiệm nhắm đến các người dùng là học sinh và môn học chủ yếu là môn toán.

+ Admin: người quản trị trong ứng dụng này là người tạo đề cho các học sinh làm bài.

##### **3.1.2. Giao diện ứng dụng**

– Admin (Giáo viên): Có 3 trang giao diện: (Phụ lục hình 1)

+ Tạo đề thi mới: Điền các thông tin như mã đề, tên đề, thời gian làm bài và chọn số lượng câu hỏi theo yêu cầu bằng cách chọn số lượng câu hỏi từng chương.

+ Điều chỉnh đề thi: Có thể Sửa Tên đề thi, thời gian làm bài Hoặc Xóa các đề thi mà học sinh chưa làm bài.

+ Điều chỉnh ngân hàng câu hỏi: Có thể Thêm, Xóa, Sửa câu hỏi trong ngân hàng đề, hỗ trợ lọc câu hỏi theo học kỳ và chương.

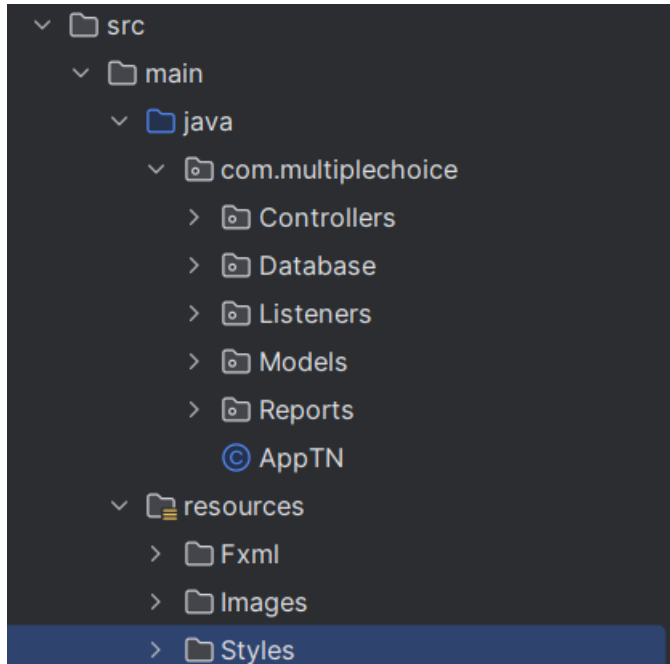
- User (Học sinh): Có 3 trang giao diện: (Phụ lục hình 3)
  - + Chọn đề muốn thi: chọn 1 trong những đề đã được giáo viên tạo từ trước.
  - + Tiến hành thi: Chọn các đáp án đúng
  - + Xem kết quả thi.

### **3.2. Cách thức hoạt động của dữ liệu qua các thành phần:**

- Người dùng tương tác với giao diện, gửi dữ liệu về controller
- Controller tương tác với mô hình dữ liệu (Thêm, xóa, sửa, lấy dữ liệu ; Cơ sở dữ liệu, tập tin, cấu trúc dữ liệu trên ứng dụng,...) và cập nhật lại giao diện người dùng, giao diện thể hiện dữ liệu đã được cập nhật.
- Ở trường hợp Ứng dụng trắc nghiệm, cách thức hoạt động sẽ bao gồm:
  - + User: Người dùng chỉ có thể chọn các đề thi và làm tại màn hình chính sau khi đăng nhập vào, các màn hình tương tác sẽ tương ứng với các controller khác nhau để xử lý các thao tác có trên giao diện, việc chọn các câu hỏi hoặc không chọn các câu hỏi sẽ được xử lý, tính toán hợp lý để (Đối với ứng dụng trắc nghiệm môn toán này) đưa dữ liệu xuống cơ sở dữ liệu có trong máy local (Nhóm không dùng kết nối với Host cơ sở dữ liệu trực tuyến) để thuận tiện cho việc lưu trữ và sử dụng sau này.
  - + Admin: Với vai trò là quản trị viên của ứng dụng thì, admin có thể:
    - Tạo đề (Lấy các đề ngẫu nhiên có trong cơ sở dữ liệu theo số lượng và học kỳ mong muốn)
    - Xem các đề thi hiện tại, điểm thi của các học sinh đã thi đề đó, tạo file DOC hoặc PDF một đề thi mà admin chọn, có thể xóa hoặc sửa đề thi nếu muốn.
    - Thêm xóa sửa các câu hỏi có trong một chương cụ thể của một học kỳ.
- Controller kiểm tra và trả lại xác nhận thành công hoặc lỗi tương ứng cho giao diện thể hiện ra cho người dùng nhận thức và tương tác với giao diện khác. Việc giao diện sử dụng các tương tác để đưa controller quản lý và việc controller xử lý các thao tác, dữ liệu đó như thế nào đều được nhóm em sử dụng mô hình MVC để thực hiện đồ án.

## 4. ỨNG DỤNG MÔ HÌNH MVC

### 4.1. Cấu trúc thư mục



Hình 4. 1: Cấu trúc thư mục

Khi sử dụng Mô hình MVC, nhóm em chia khái niệm – tương đương với các tập tin thành các thư mục theo mô hình để tiện cho việc quản lý cũng như bảo trì. Source code sẽ có thư mục “java” chứa tất cả các tập tin có đuôi là .java bao gồm thư mục:

- + Models chứa các tập tin về các thực thể trong cơ sở dữ liệu quan hệ (Thư mục này đảm nhận phần **Model** trong mô hình MVC)
- + Thư mục “resources” với mục đích là chứa các tập tin liên quan đến giao diện và bố trí giao diện (Thư mục này đảm nhận phần **View** trong mô hình MVC).
- + Controllers (Thư mục này chứa các tập tin liên quan đến xử lý dữ liệu và tương tác giao diện, đảm nhận phần **Controller** trong mô hình MVC).

### 4.2. Chi tiết ứng dụng với mô hình MVC

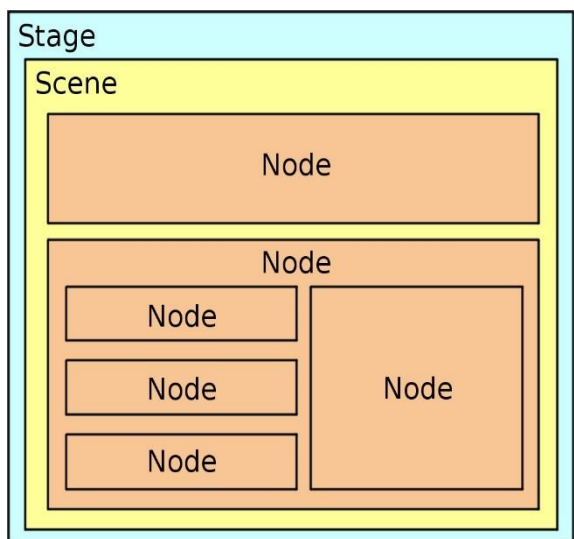
#### 4.2.1. View:

Phần giao diện của nhóm sẽ bao gồm các trang lớn nhỏ khác nhau.

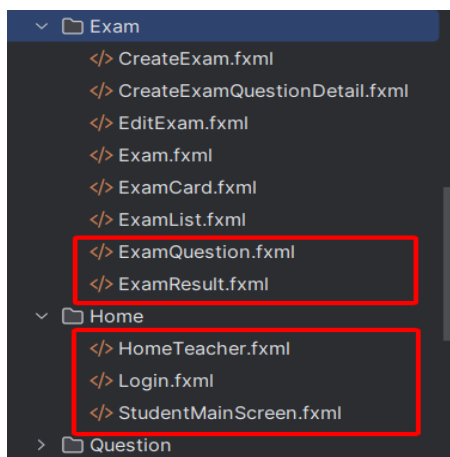
Các tập tin có đuôi .fxml chứa các node đều có thể là một node trong một trang khác. Các trang lớn bao gồm: trang login (Login.fxml), trang chủ của người dùng thông thường (StudentMainScreen.fxml), trang thi cử (ExamQuestion.fxml), trang kết quả thi



(ExamResult.fxml), trang chủ của admin (HomeTeacher.fxml)

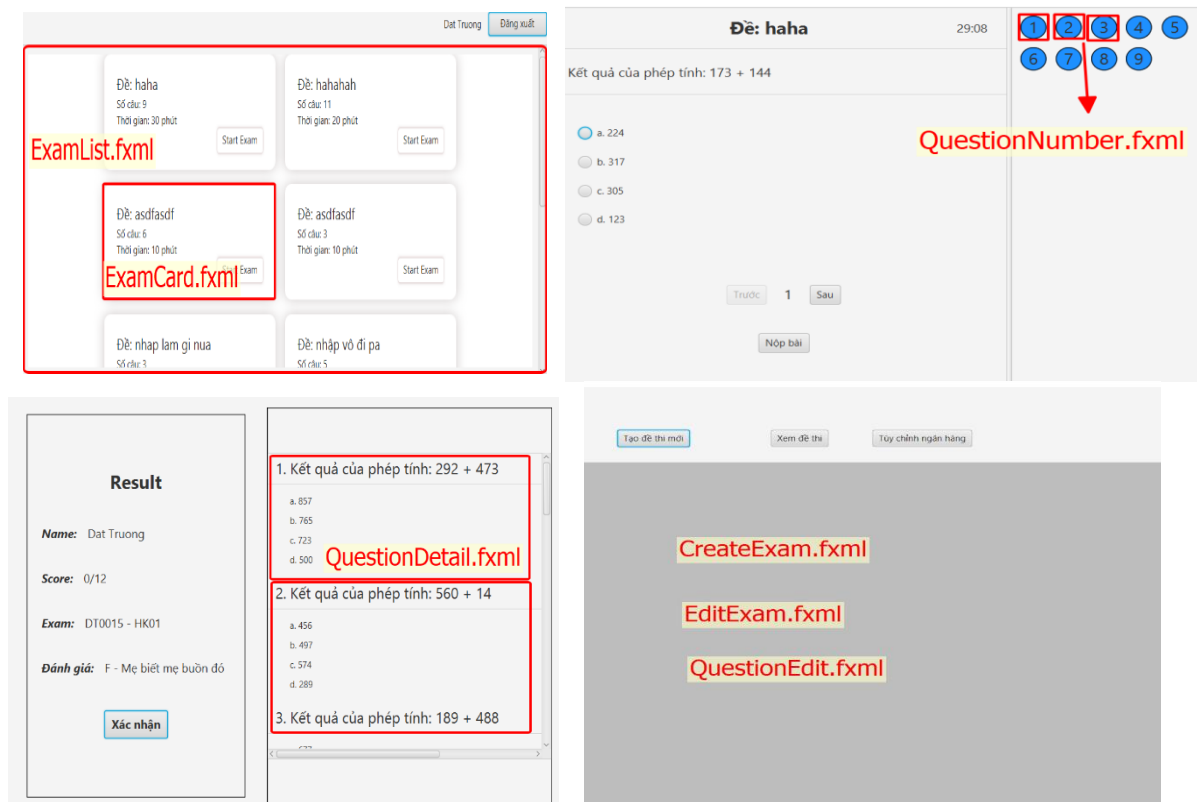


Hình 4. 2: JavaFX - Cấu trúc tổng quát của các thành phần

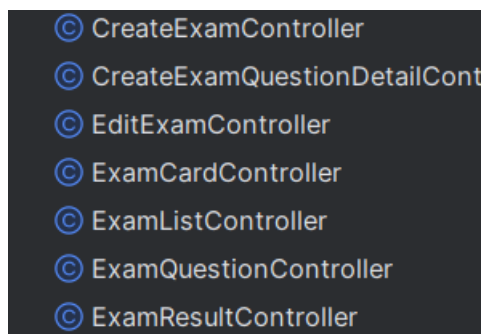


Tất cả các tập tin còn lại đều là các trang giao diện nhỏ được thêm vào các node lớn để xử lý thành các giao diện hoàn chỉnh

Hình 4. 3: Các trang giao diện lớn



Hình 4. 4: Các trang nhỏ



Hình 4. 5: Các class Controller

### 4.2.2. 4.2.2. Controller:

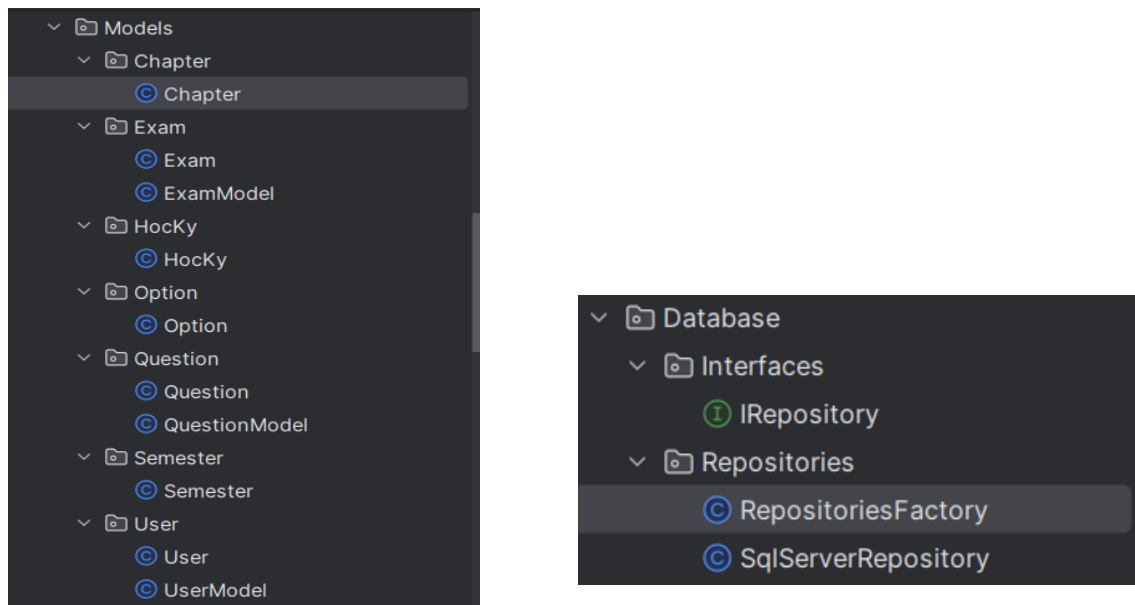
Vì phần View của JavaFX luôn cần một Controller để xử lý nên để tiện xử lý thì nhóm em đặt tên các Controller trùng với tên của các tập tin có đuôi .fxml.

Các Controllers cần sử dụng các Models để thực hiện xử lý dữ liệu được truyền xuống từ Views và các thao tác lấy dữ liệu từ cơ sở dữ liệu để thể hiện lên Views.

### 4.2.3. 4.2.3. Model:

- Các tập tin trong Models tượng trưng cho các thực thể chính có trong cơ sở dữ liệu.

- Ngoài ra còn có các tập tin như QuestionModel, UserModel, ExamModel. Đây là các class dùng để gọi tới Database thông qua việc gọi tới class kết nối tới cơ sở dữ liệu: SqlServerRepository.



Hình 4. 6: Các Model và Class Kết nối Database

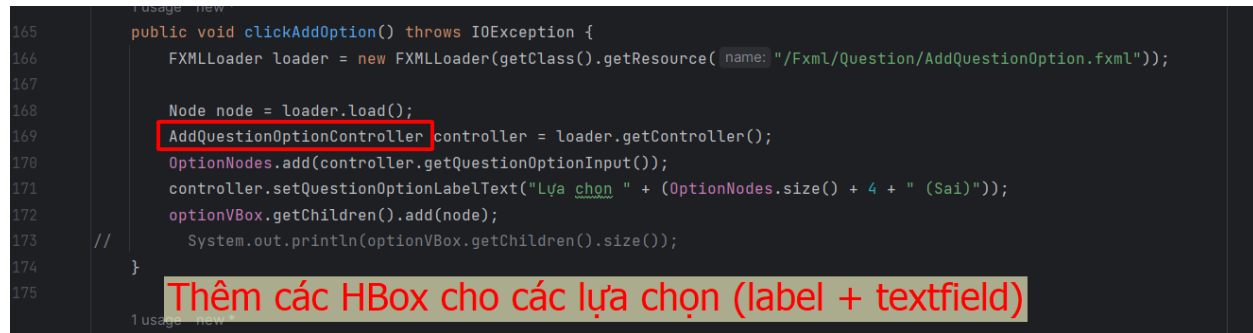
## 5. BỔ SUNG MỞ RỘNG

### 5.1. Thêm lựa chọn sai

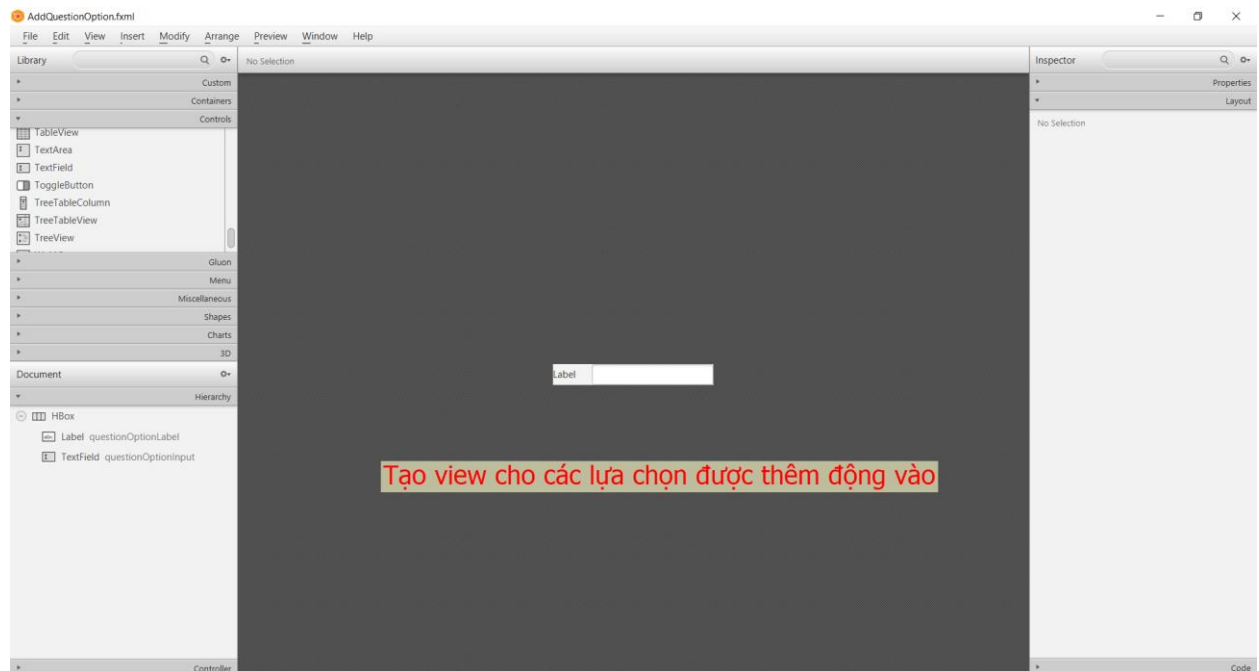
Thêm các lựa chọn sai nhưng khi hiển thị chỉ chọn ngẫu nhiên hiển thị 1 bộ gồm (1 lựa chọn đúng + 3 lựa chọn sai)

#### 5.1.1. Phần view

Tạo HBox để chứa các lựa chọn thêm vào: Label và textfield



Tạo view cho các lựa chọn được thêm động vào, sẽ được thêm khi lick chọn thêm option sẽ thêm vào trong Hbox



Bổ xung thêm phần lấy thông tin ở các phần option mới thêm vào

```
84     public void clickAddQuestion() throws Exception {
85         alert.setHeaderText("Thông báo");
86         if(!checkInput()) {
87             alert.setContentText("Điền đầy đủ thông tin vào Nội dung và các ô Lựa chọn");
88             alert.showAndWait();
89             return;
90         }
91         try {
92             ArrayList<String> options = new ArrayList<>(Arrays.asList(txfA.getText(), txfB.getText(), txfC.getText(), txfD.getText()));
93             addToOptions();
94             for(String option : OptionList) {
95                 options.add(option);
96             }
97             QuestionModel.add(txfIdQuestion.getText(), txfIdChapter.getText(), txaContent.getText(), options);
98         } catch (ClassNotFoundException | SQLException e) {
99             // TODO Auto-generated catch block
100             System.out.println(e);
101             alert.setContentText("Không thêm được câu hỏi!");
102             alert.showAndWait();
103             System.out.println(e);
104             // e.printStackTrace();
105         }
106     }
107 }
```

OptionList: giá trị của các textfield mới

options: Giá trị của tất cả các textfield

Giao diện thêm các lựa chọn sai

Ứng dụng quản lý trắc nghiệm

ThaySon Đăng xuất

Tạo đề thi mới Ngân hàng đề thi Ngân hàng câu hỏi

Học kỳ 1 Học kỳ 2

Mã chương	Tên chương
CH01	Phép cộng trong phạm vi 1000
CH02	Phép trừ trong phạm vi 1000
CH03	Phép nhân trong phạm vi 1000
CH04	Phép chia trong phạm vi 1000
CH05	Hình học + Đơn vị đo
CH06	Tính giá trị biểu thức trong phạm vi 1000

Mã chương CH01  
Tên chương Label  
Mã câu hỏi  
Nội dung câu hỏi

Lựa chọn 2 (Sai)    
Lựa chọn 3 (Sai)    
Lựa chọn 4 (Sai)    
Lựa chọn 5 (Sai)    
Lựa chọn 6 (Sai)

Xóa Thêm Sửa Thêm lựa chọn

AddQuestionOption.fxml

Mã câu hỏi	Nội dung
------------	----------

### 5.1.2. Phần mã nguồn

Tạo hàm trả về mảng các lựa chọn. Tạo 2 Query để truy vấn

Query 1: Lấy đáp án đúng

Query 2: Lấy ngẫu nhiên 3 lựa chọn sai (dùng select top3 và order by NEWID())

Thêm giá trị truy vấn được vào mảng option.

```

3 usages  unknown *
private static Option[] getOptionsFromQuestion(Question qu) {
    Option[] options = new Option[4];
    String correctOptionQuery = String.format(
        "SELECT questionoption.idoption, questionoption.title, questionoption.isCorrect " +
        "FROM questionoption " +
        "WHERE questionoption.idquestion = '%s' AND questionoption.isCorrect = 1",
        qu.getId().replaceAll(regex: "[^a-zA-Z0-9]", replacement: "")
    );
    String incorrectOptionQuery = String.format(
        "SELECT TOP 3 questionoption.idoption, questionoption.title, questionoption.isCorrect " +
        "FROM questionoption " +
        "WHERE questionoption.idquestion = '%s' AND isCorrect = 0 " +
        "ORDER BY NEWID()",
        qu.getId().replaceAll(regex: "[^a-zA-Z0-9]", replacement: "")
    );
};

try {
    ResultSet rs = repo.executeQuery(correctOptionQuery);
    if(rs.next()) {
        Option option = new Option();
        option.setId(rs.getString(columnLabel: "idoption"));
        option.setTitle(rs.getString(columnLabel: "title"));
        option.setIsCorrect(rs.getBoolean(columnLabel: "isCorrect"));
        options[0] = option;
    }

    rs = repo.executeQuery(incorrectOptionQuery);
    int index = 1;
    while(rs.next()) {
        Option o = new Option();
    }
}

```

Ngẫu nhiên 3 đáp án sai

Lấy đáp án đúng

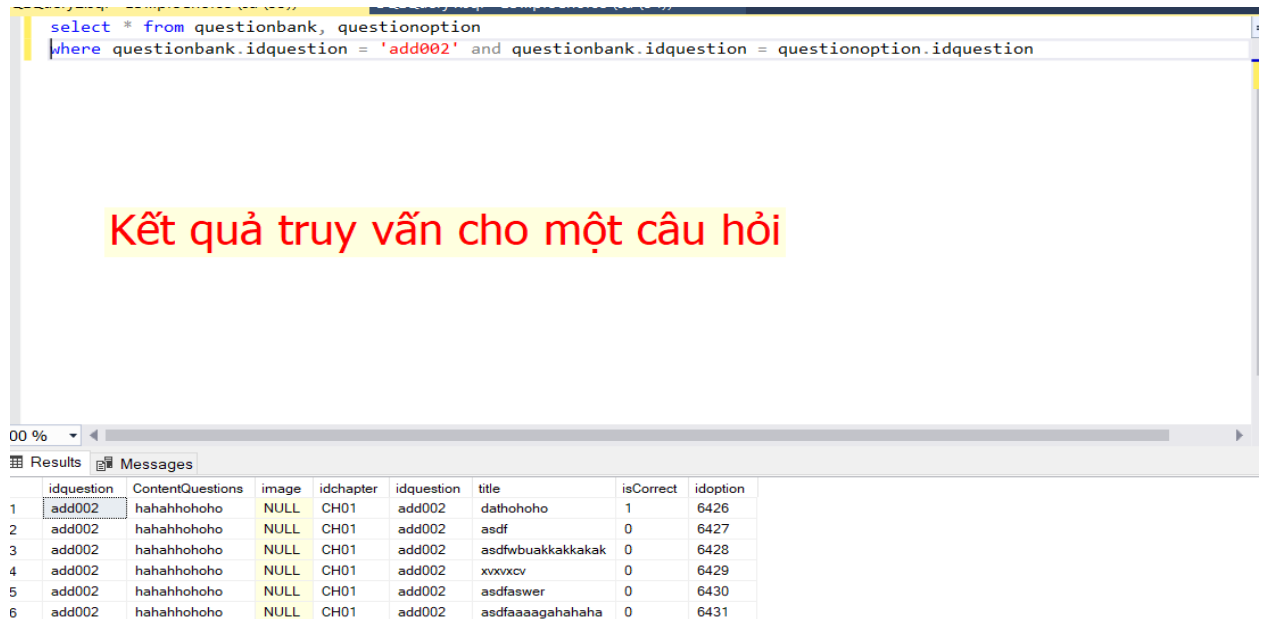
Lấy các đáp án sai

Phản hiển thị các đáp án dùng phương thức shuffle của lớp Collections để thực hiện việc xáo trộn dữ liệu: truyền vào mảng các lựa chọn và đảo ngẫu nhiên.

```

public void setQuestionToggleGroup(Option[]
questionOptions, ToggleGroup currentQuestionGroup) {
    List<Option> shuffledQuestionOptions =
Arrays.asList(questionOptions);
    Collections.shuffle(shuffledQuestionOptions);
    //...hiển thị đáp án vào view
}

```



```
select * from questionbank, questionoption
where questionbank.idquestion = 'add002' and questionbank.idquestion = questionoption.idquestion
```

**Kết quả truy vấn cho một câu hỏi**

	idquestion	ContentQuestions	image	idchapter	idquestion	title	isCorrect	idoption
1	add002	hahahhohoho	NULL	CH01	add002	dathohoho	1	6426
2	add002	hahahhohoho	NULL	CH01	add002	asdf	0	6427
3	add002	hahahhohoho	NULL	CH01	add002	asdfwbuakkakkakak	0	6428
4	add002	hahahhohoho	NULL	CH01	add002	xxxxxcv	0	6429
5	add002	hahahhohoho	NULL	CH01	add002	asdfaswer	0	6430
6	add002	hahahhohoho	NULL	CH01	add002	asdfaaaagahahaha	0	6431

## 5.2. Giải thích cơ chế lấy ngẫu nhiên câu hỏi và đảo đáp án

### 5.2.1. Lấy ngẫu nhiên câu hỏi

Dùng cơ chế sắp xếp ngẫu nhiên của SQL server

`ORDER BY NEWID()` là một cơ chế trong SQL Server được sử dụng để sắp xếp kết quả truy vấn theo thứ tự ngẫu nhiên.

Cơ chế `ORDER BY NEWID()` hoạt động như sau:

1. SQL Server tạo một cột ảo (virtual column) mới có tên là `NEWID()`. Hàm `NEWID()` tạo một giá trị GUID (Globally Unique Identifier) ngẫu nhiên mỗi khi nó được gọi.
2. SQL Server sử dụng giá trị GUID được tạo từ `NEWID()` để sắp xếp các hàng trong kết quả truy vấn. Vì GUID là duy nhất và ngẫu nhiên, nên việc sắp xếp theo `NEWID()` sẽ tạo ra một thứ tự ngẫu nhiên. Ví dụ

```
SELECT * FROM Question ORDER BY NEWID();
```

Trong ví dụ này, bảng `Question` sẽ được sắp xếp theo một thứ tự ngẫu nhiên.

Tuy nhiên cơ chế này có thể gây tốn tài nguyên máy chủ khi áp dụng cho các bảng lớn, vì mỗi hàng trong bảng cần phải có một giá trị GUID ngẫu nhiên được tạo và sắp xếp.

### 5.2.2. Đảo đáp án

Phương thức shuffle trong Java (của thư viện java.util.Collections) được sử dụng để xáo trộn (hoặc ngẫu nhiên sắp xếp lại) các phần tử trong một danh sách (List) hoặc mảng (Array). Cơ chế của phương thức shuffle đơn giản và hiệu quả. Cách sử dụng phương thức `shuffle`:

1. Import thư viện Collections: `import java.util.Collections;`
2. Tạo một danh sách (List) hoặc mảng (Array) cần xáo trộn.
3. Sử dụng phương thức `shuffle` để xáo trộn danh sách (List):

Ví dụ `Integer[] myArray = {1, 2, 3, 4, 5};`

```
List<Integer> myList = Arrays.asList(myArray);  
Collections.shuffle(myList);
```

Sau khi thực hiện phương thức `shuffle`, các phần tử trong danh sách `myList` sẽ được xáo trộn một cách ngẫu nhiên.

Việc xáo trộn này là cách xáo trộn ngẫu nhiên, và không phụ thuộc vào giá trị của các phần tử trong danh sách.

### 5.3. Xuất file PDF

**Lưu ý:** Jasper Report hiện chỉ đang hỗ trợ trên các IDE như **NetBeans**, **Eclipse**

Các bước thực hiện:

The screenshot displays the Jasper Report software interface. On the left, there's a 'Form Designer' area with a table of data. The table has columns: 'Học kỳ', 'Mã đề', 'Tên đề', 'Số câu', and 'Thời gian làm bài'. The data rows are as follows:

Học kỳ	Mã đề	Tên đề	Số câu	Thời gian làm bài
HK01	DT01045	A	2	1
HK01	DT010	Nhập với đề pa	1	10
HK01	DT01003	suoi	1	10
HK02	DT01001	De KT 15P	3	1
HK02	DT01002	De KT 15P	4	1
HK01	DT01000	Ai thì ra đây	11	10

On the right, there's a 'Preview' area showing a PDF report titled 'Đề thi - De KT 15P'. The report contains 4 multiple-choice questions. A 'Save' dialog box is open over the PDF preview, with the 'Save' button highlighted.



## 6.KẾT LUẬN

Tóm tắt lại quá trình thực hiện:

- Phác thảo mô hình ứng dụng.
- Tạo cơ sở dữ liệu trên SQL server.
- Thiết kế giao diện ứng dụng bằng JavaFX và Scene Builder.
- Kết nối với cơ sở dữ liệu SQL Server với thư viện JDBC.
- Tạo logic xử lý dữ liệu và Thiết kế các chức năng.
- Kiểm tra và sửa lỗi.

Kết quả đạt được:

- Kết nối, tương tác được với CSDL SQL server
- Tạo giao diện ứng dụng và kết nối được các Controller điều khiển
- Thực thi các lệnh thêm, xóa, sửa Đề Thi, Ngân hàng Câu hỏi, ngân hàng câu trả lời một cách dễ dàng, trực quan với người dùng.
- Hiện thị đề thi trực quan, thi trực tiếp với thời gian giới hạn từ trước, tính điểm và lưu trữ kết quả thi.
- Xuất được Đề Thi dạng PDF.

Mở rộng: Hiện tại nhóm chỉ xử lý được từng dòng dữ liệu, trong tương lai nhóm muốn phát triển ứng dụng có thể thêm, xóa, sửa hàng loạt để mở rộng, điều chỉnh CSDL nhanh chóng, hiệu quả hơn.

## TÀI LIỆU THAM KHẢO

- [1] Carl Dea, Mark Heckler, Gerrit Grunwald, Jose Pereda Ph. D, Sean M Phillips...JavaFX 8 Introduction by Example 2<sup>nd</sup> Edition, 2014.
- [2] Brady Ellison, Java for Beginners: A Crash Course to Learn Java Programming in 1 Week, 2022.
- [3] Hà Huy Khoái, Lê Anh Vinh...Toán 3 tập 1 Nhà xuất bản Giáo dục Việt Nam, 2022.
- [4] Hà Huy Khoái, Lê Anh Vinh...Toán 3 tập 2 Nhà xuất bản Giáo dục Việt Nam, 2022.

**PHỤ LỤC PHÂN CÔNG NHIỆM VỤ**

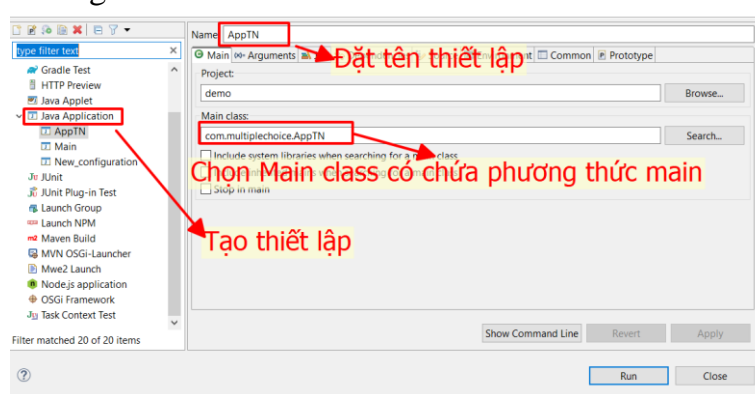
STT	Thành viên	Nhiệm vụ
1	Trương Quang Đạt	Thiết kế mô hình ứng dụng, tạo mã nguồn 50%, tạo CSDL, thiết kế giao diện, viết word, viết powerpoint
2	Đinh Hoàng Thùy Linh	Thiết kế mô hình ứng dụng, tạo mã nguồn 50%, tạo CSDL, thiết kế giao diện, viết word, viết powerpoint

## PHỤ LỤC DEMO

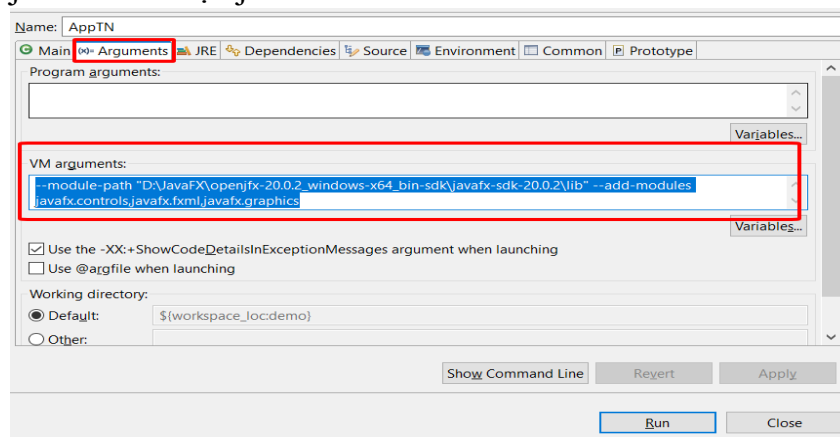
Chạy sản phẩm theo các bước sau:

### Bước 1:

- Click vào đường dẫn tới github và clone về máy thư mục source code ứng dụng trắc nghiệm. Đường dẫn: <https://github.com/d1nnn/multiplechoiceapp>
- Mở thư mục bằng Eclipse hoặc NetBeans (hoặc IDE khác – khuyến khích nên sử dụng 2 IDE này vì thư viện Jasper Studio chỉ hỗ trợ trên những IDE này).
- Đối với Eclipse, nếu ứng dụng chưa thể tự tải dependencies, mở thư mục demo sẽ thấy một tập tin là: pom.xml. Click chuột phải -> Chọn Maven -> Chọn Update Project -> Đợi tải và build.
- Sau khi build xong, nhấp chuột phải vào project và chọn Run as -> Run Configurations...



- Chọn tab Arguments và đưa vào VM arguments dòng lệnh: **--module-path "Đường/Dẫn/Đến/lib" --add-modules javafx.controls,javafx.fxml,javafx.graphics**, ở đây thư mục lib chứa các file jar cho thư viện javafx.

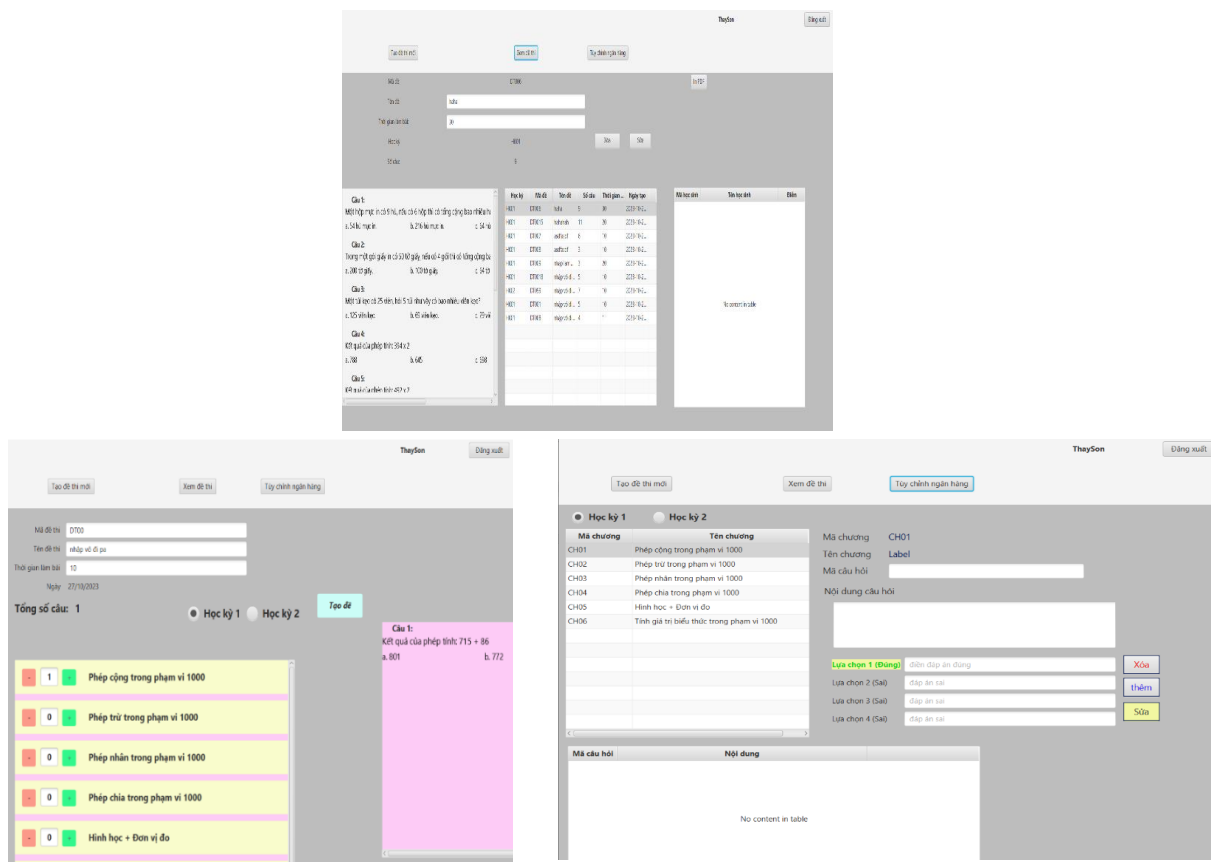


**Bước 2:** Tạo cơ sở dữ liệu như đã giới thiệu. Tạo hàng trong tblUser là admin (isAdmin) và user.

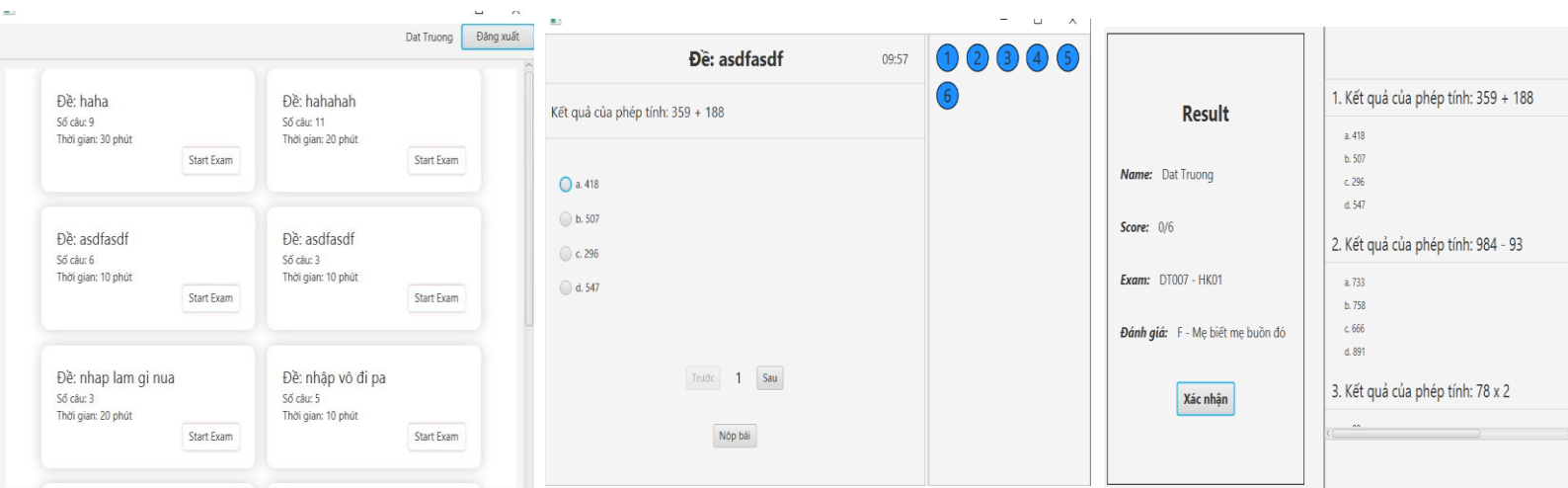
**Bước 3:** Thiết lập driver JDBC cho DriverManager (username, password,... )

**Bước 4:** Chạy ứng dụng và đăng nhập.

## PHỤ LỤC HÌNH



Hình 1: Giao diện tổng quát cho admin



Hình 2: Giao diện tổng quát User