Fpt University

TRƯỜNG ĐẠI HỌC FPT

CAPSTONE PROJECT REPORT

Report 4 – Software Design Document

ONLINE LEARNING SYSTEM

– Hanoi, June 26, 2023 –

I. Project Report

1. Status Report

#	Work Item	Status	Notes (Work Item in Details)
1	Overall Description	Completed	
2	System Architecture Design	Completed	
3	System Detailed Design	Completed	
4	Object Specifications	Completed	
5	Data & Database Design	Completed	

2. Team Involvements

#	Task	Member	Notes (Task Details, etc.)
1	Overall Description	DungNM,NguyenTK	
2	System Architecture Design	KhaNM,ThaiLA,HưngNM	
3	System Detailed Design	KhaNM,ThaiLA,HưngNM	
4	Object Specifications	DungNM,NguyenTK	
5	Data & Database Design	KhaNM,ThaiLA,HưngNM	

3. Issues/Suggestions

#	Issue	Status	Notes (Solution, Suggestion, etc.)
1		Pending	
2		In Progress	
3		Completed	

II. Software Design Document

1. Overall Description

1.1 Assumptions

This system is designed based on the following assumptions:

- Window 11
- Spring boot
- MySQL
- Google Chrome

1.2 Design Constraints

This system should comply with the following items:

- It has 4 websites at the client-side to serve two types of actors: guest, student, staff and admin. They can use a web browser and application to send requests and connect to the front-end application.
- End-user's Environment: Windows.
- Database using MySQL.
- Support languages: English.
- The user must have a stable connection to the internet.

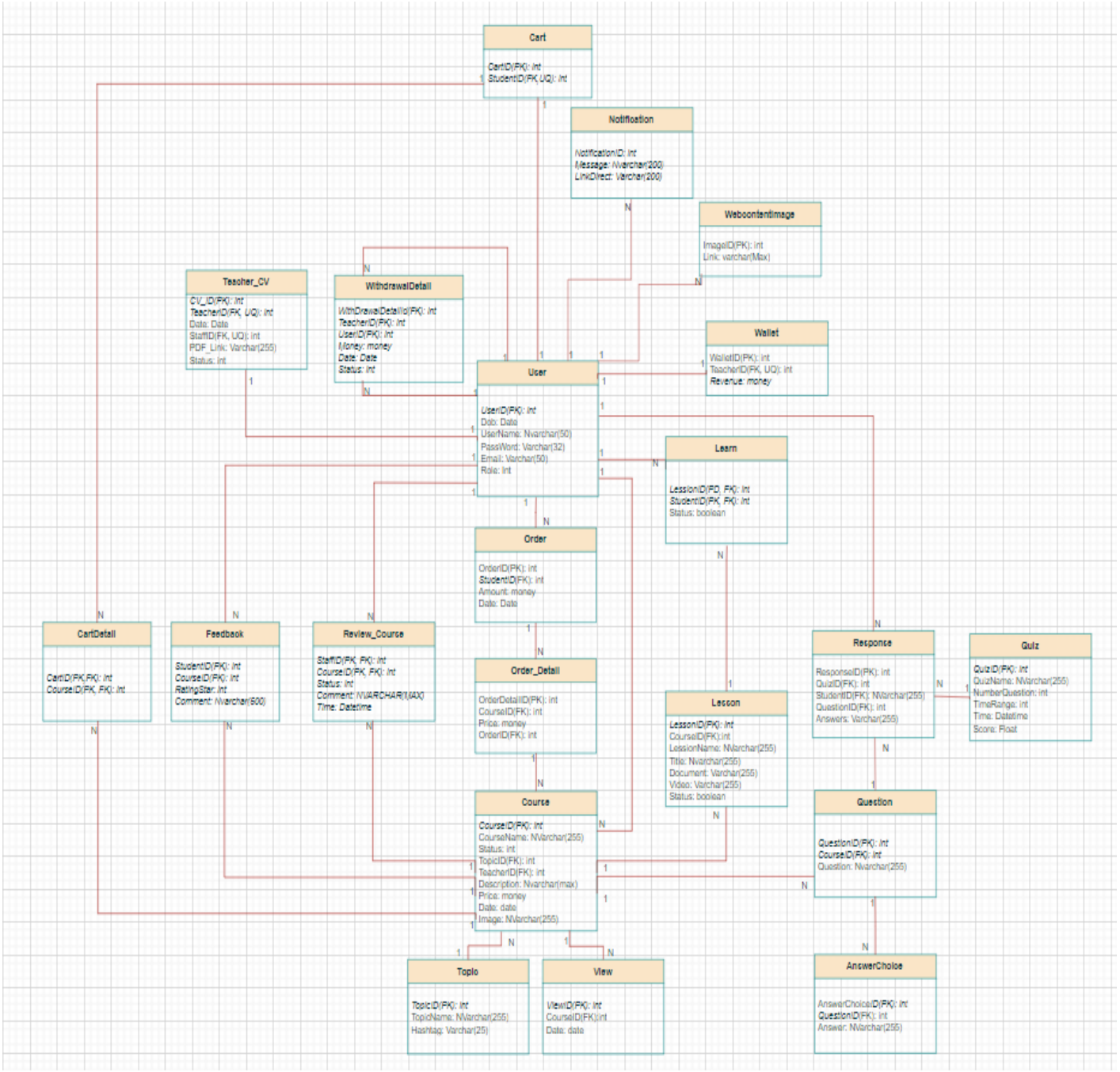
1.3 Technology Suggestion

- Database: MySQL
- Web Application:, Tomcat 8.5.3
- Server: Local server

2. System Architecture Design

2.1 Overall Architecture

2.1.1 Diagram



2.2.2 Component Explanation

2.2.2.1 Bootstrap

Bootstrap 5 is a frontend framework and development platform for creating efficient and sophisticated web applications. With Bootstrap 5, you can leverage a scalable platform for projects ranging from small endeavors to large-scale enterprise applications. Bootstrap 5 is designed to make updating easy, allowing you to take advantage of the latest developments with minimal effort. Best of all, the Bootstrap ecosystem consists of a diverse community of over 1.7 million developers, library authors, and content creators.

2.2.2.2 MySQL

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

2.2.2.3 Spring Framework

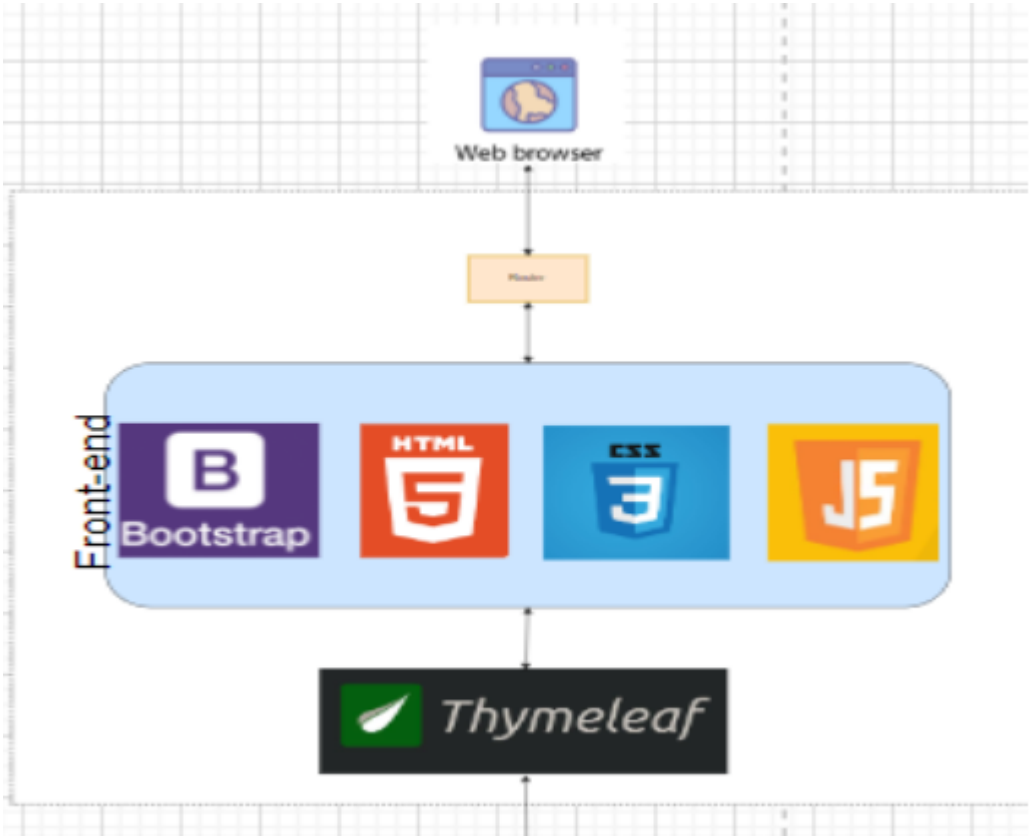
The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to the Enterprise Javabeans (EJB) model. The Spring Framework is open source.

2.2.2.4 Thymeleaf

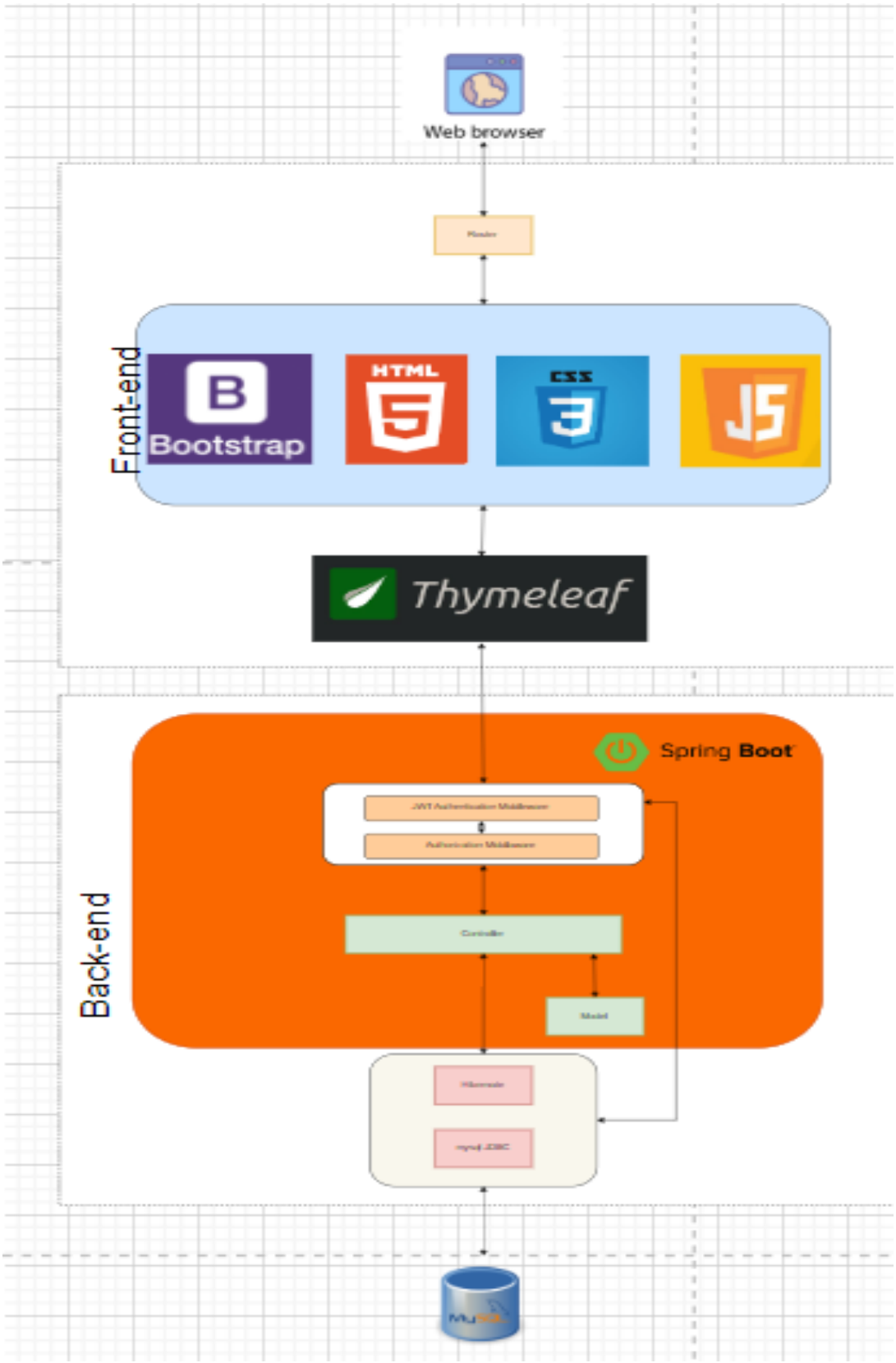
Thymeleaf is a popular template engine and frontend development framework in the Java environment. It is designed to create dynamic and interactive web pages by combining Java code with HTML templates. Thymeleaf features a user-friendly syntax and supports various functionalities such as expressions, conditionals, loops, form creation, and internationalization. Its key strength lies in its seamless integration with Java frameworks like Spring Boot, making it efficient for building effective and easily maintainable web applications on both the server and client-side.

2.2 System Architecture

2.2.1 Onvid Front-end Architecture

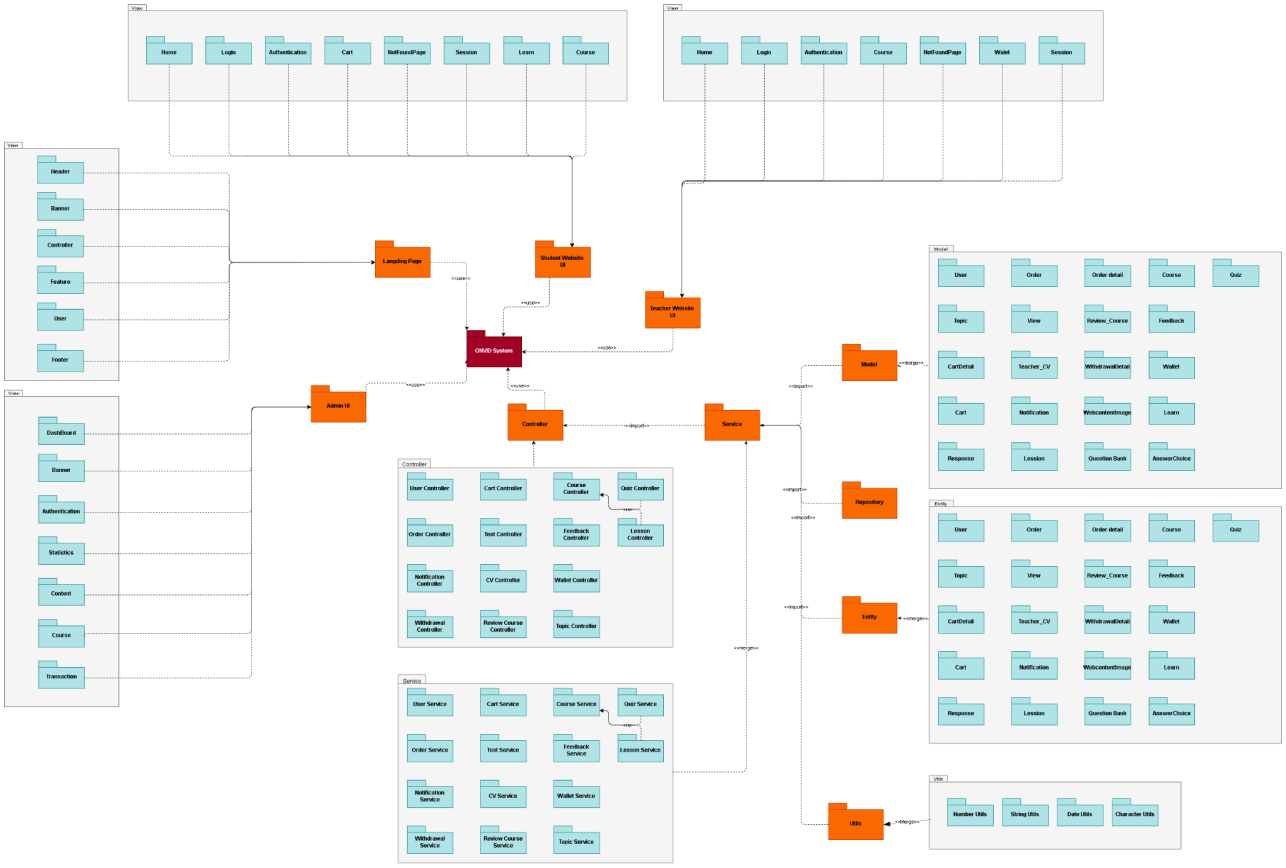


2.2.2 Onvid Back-end Architecture



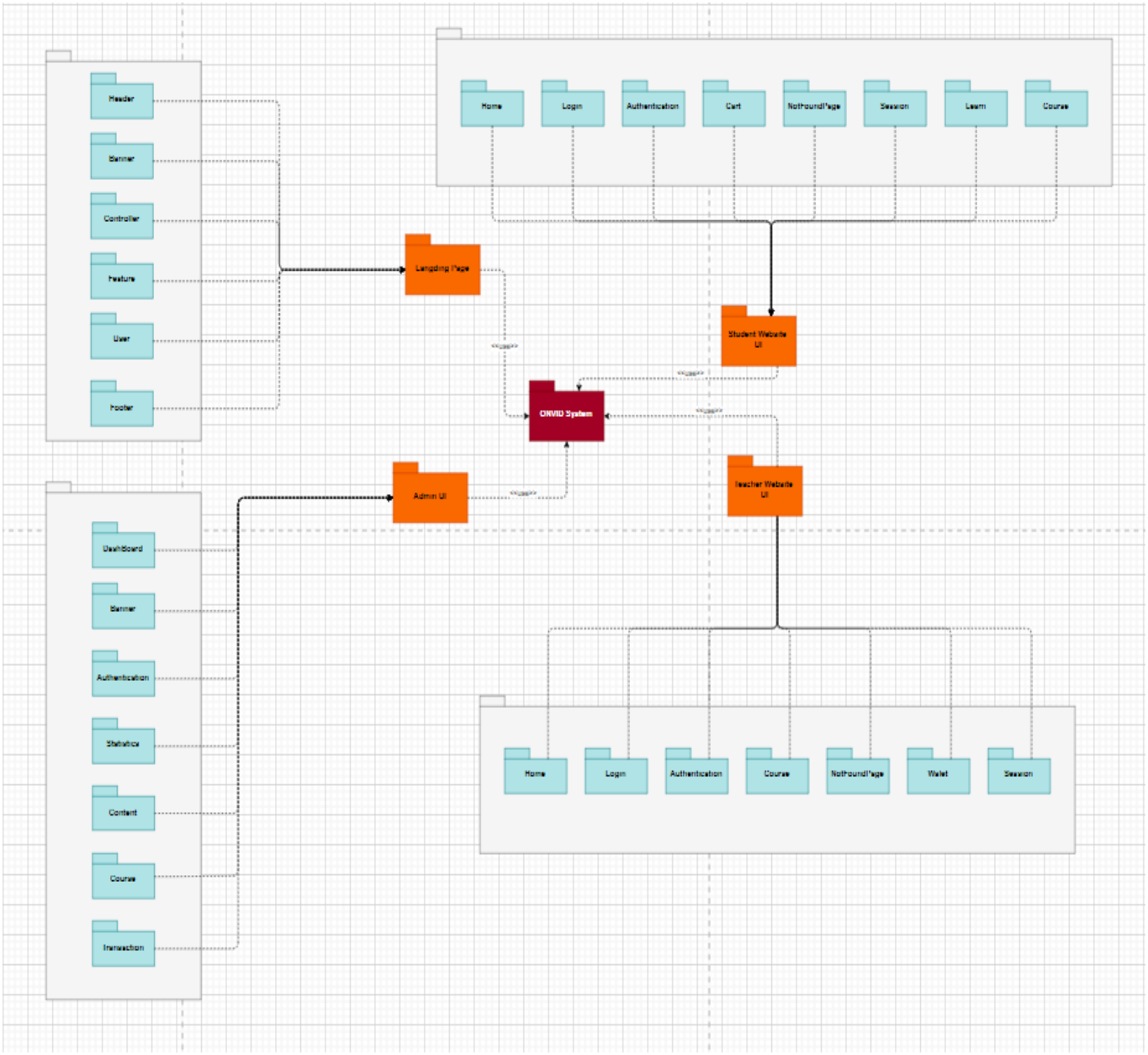
2.3 Package Diagram

2.3.1 Onvid Package Diagram



2.3.2 OnVid Front-end

2.3.2.1 Diagram

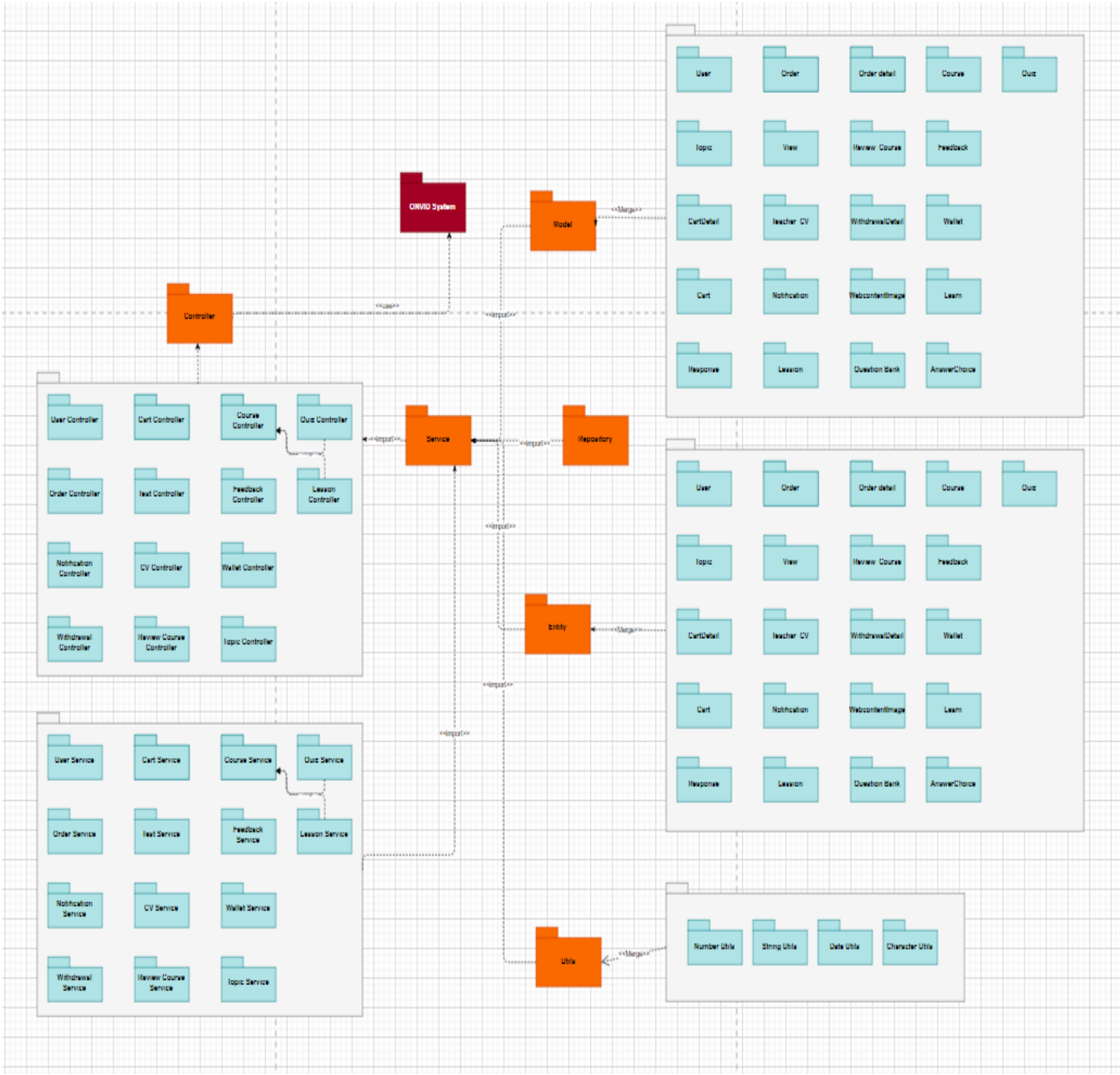


2.3.2.2 Package Explanation

No	Package	Description
1	ONVID System	Contain all code of ONVID system.
2	Student Website UI	Contain all client code for the main website.
3	Landing Page	Contain all code of a website for the first time.
4	Admin UI	Contain all client code for the managed website.
5	Teacher Website UI	Contain all client code for the main website.

2.3.3 OnVid Back-end

2.3.3.1 Diagram



2.3.3.2 Package Explanation

No	Package	Description
1	Controller	receive the request from the user and assign the work to the service layer.
2	Model	contains the DTO used to return the necessary data to the client
3	Repository	connect to database.
4	Service	receive the request from the controller and assign the work to the repository layer
5	Entity	define fields and constraints between classes
4	Utils	Contains frequently used functions

3. Class Specifications

3.1 Controller

3.1.1 Student

3.1.1.1 CartController

No	Method	Description
1	addToCart	This action is used to add a course to the user's shopping cart. When a user finds a course they want to enroll in, they can click on the "Add to Cart" button, and this action will be triggered to place the course in their cart.
2	cartDetail	This action is used to view the details of the items present in the user's shopping cart. It allows the user to see a list of courses they have added to the cart along with relevant information such as course names, prices, and any applicable discounts.
3	deleteCourseInCart	This action allows the user to remove a specific course from their shopping cart. If the user changes their mind or no longer wants to purchase a particular course, they can use this action to delete it from the cart.
4	deleteAllCoursInCart	This action is used to clear the entire shopping cart in one go. If the user wants to remove all the courses they have added to the cart and start over, they can trigger this action to delete all the items in the cart.

3.1.1.2 LearnController

No	Method	Description
1	setLearnDefault	This action likely refers to setting the default learning preferences or settings for a user.
2	changeLearnStatus	This action is used to change the learning status of a user for a particular course or learning module

3.1.1.3 DoTestController

No	Method	Description
1	getTestOfCourse	This action is likely used to retrieve or fetch the test or assessment associated with a specific course.
2	postMethodName	This action appears to be a placeholder or a generic name rather than a specific action.

3.1.1.4 CourseController

No	Method	Description
1	detạil	This action likely refers to fetching detailed information about a specific course.
2	getMyCourse	This action is used to retrieve the list of courses that the currently authenticated user has enrolled in or purchased.

3.1.2 Teacher

3.1.2.1 CourseController

No	Method	Description
1	getCourseList	This action is used to fetch a list of available courses on the online platform. When a user wants to explore the courses offered, they can trigger this action, and the server will respond with a list of courses, including their names, descriptions, instructors, and other relevant details.
2	getCreateCourse	This action is a bit ambiguous, but it likely refers to retrieving the necessary information or form for creating a new course.
3	addCourse	This action is used to create a new course on the platform. When an instructor or administrator has filled out the necessary details for a new course

4	getUpdateCourse	they can submit the data using this action to add the course to the platform's database
5	postUpdateCourse	This action is used to update an existing course on the platform. After making the necessary changes to a course an instructor or administrator can use this action to submit the updated data to the server, which will then apply the changes to the course in the database.
6	getDeleteCourse	used to initiate the process of deleting a course from the online platform. When an instructor or administrator wants to remove a course from the platform, they can trigger this action to start the deletion process.
7	getDetail	used to retrieve detailed information about a specific course. Similar to the detail action mentioned in a previous response, it allows users to view comprehensive information about a course, such as its name, description, instructor details, curriculum, pricing, and any other relevant data.
8	getSubmitCourse	This action could be related to course creation or updating. It might be used to submit a new course or updated course data to the server for processing and storage. F
9	getReview	This action likely refers to accessing a course review or feedback page. Users, typically learners who have completed or engaged with a course, can access this action to leave a review or provide feedback about the course they took.
10	getFeedback	specifically focus on receiving feedback from users about the overall learning experience or platform features. It could be a more general feedback form for users to express their thoughts and suggestions about the platform as a whole.

3.1.2.2 QuestionController

No	Method	Description
1	showAddQuestionForm	used to display a form for adding a new question. When an administrator or authorized user wants to create a new question for an assessment or quiz, they can trigger this action. The form will typically include fields to enter the question, answer options, correct answer, and any other relevant information.
2	addQuestion	used to add a new question to the platform's database or storage.
3	getDeleteQuestion	used to initiate the process of deleting a question from the platform. When an administrator or authorized user wants to remove a question from the question bank or a specific assessment, they can trigger this action., it might prompt for additional confirmation before actually removing the question permanently.
4	getDetailQuestion	used to retrieve detailed information about a specific question.

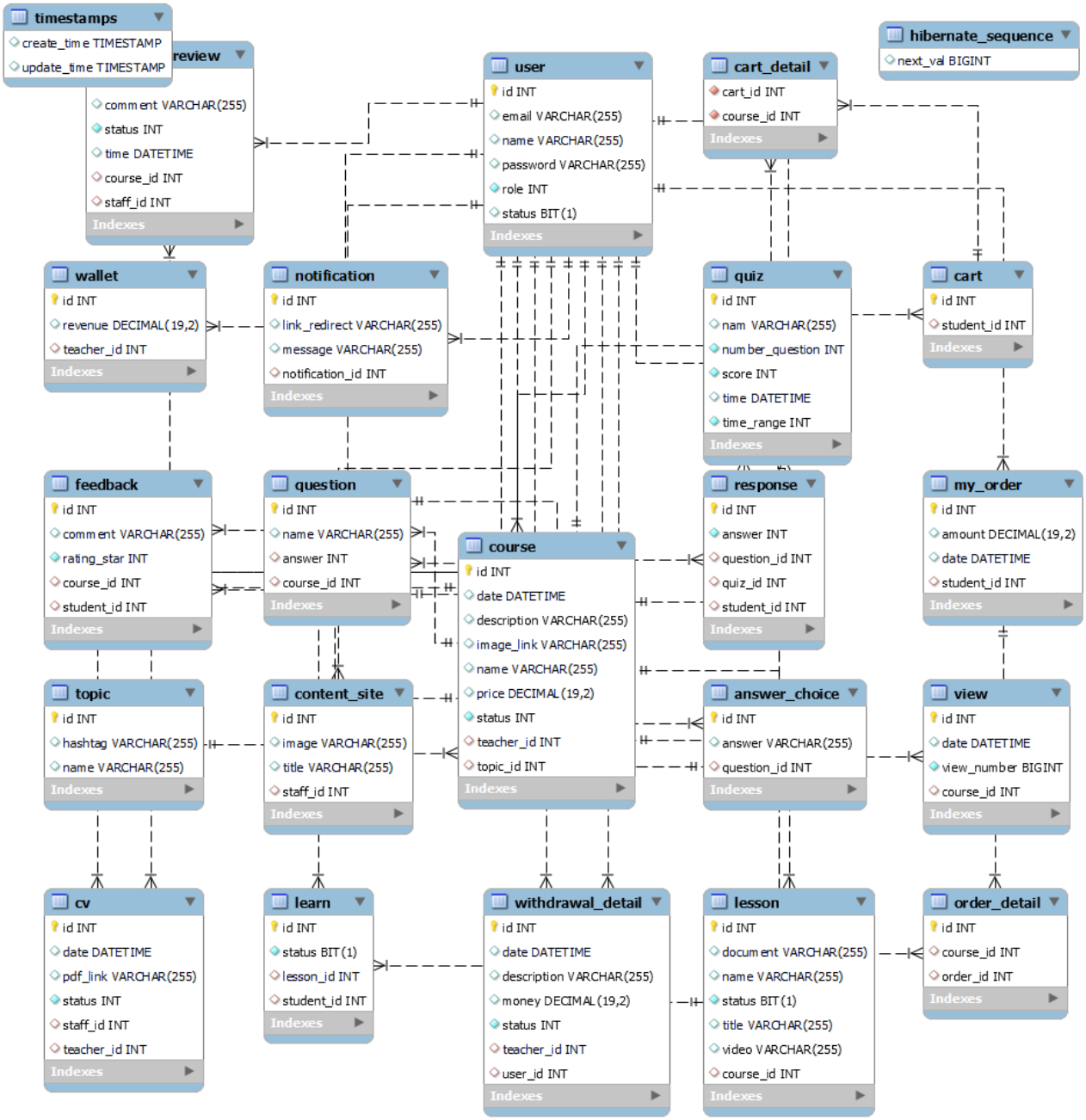
5	postUpdateQuestion	used to update an existing question with new information

3.1.2.3 WithdrawalDetailController

No	Method	Description
1	getListWallet	used to fetch a list of wallets associated with a user or an account on the platform.
2	getCreateWallet	used to retrieve the necessary information or form for creating a new wallet.
3	postCreateWallet	used to create a new wallet on the platform. After filling out the necessary details for the new wallet

4. Data & Database Design

4.1 Database Design



4.1.1. AnswerChoice

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	AnswerChoice*ID*	*int*	Y	N	PK	Auto-generation id.
2	*QuestionID*	*int*	N	Y	FK	Question of answer.
3	Answer	String	N	N		Answer details.

4.1.2. Question

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*QuestionID*	*int*	Y	N	**PK**	Auto-generation id.
2	*CourseID*	*int*	N	N	**FK**	
3	Question	String	N	Y		

4.1.3. Response

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	ResponseID	int	Y	N	**PK**	Auto-generation id.
2	QuizID	int	N	N	**FK**	
3	StudentID	int	N	N	**FK**	
4	QuestionID	int	N	N	**FK**	
5	Answers	String	N	Y		

4.1.4. Quiz

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*QuizID*	*int*	Y	N	**PK**	Auto-generation id.
2	QuizName	string	N	N		Account id of a log.
3	NumberQuestion	int	N	N		Action type of a log.
4	TimeRange	int	N	N		Note message of a log.
5	Time	Datetime	N	N		Date time of the created log.
6	Score	Float	N	N		

4.1.5. View

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*ViewID*	*int*	Y	N	**PK**	Auto-generation id.
2	CourseID	*int*	N	N	**FK**	
3	Date	*int*	N	N		

4.1.6. Topic

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*TopicID*	*int*	Y	N	**PK**	Auto-generation id.
2	TopicName	*String*	N	N		
3	Hashtag	*String*	N	N		

4.1.7. Course

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*CourseID*	*int*	Y	N	**PK**	Auto-generation id.
2	CourseName	*String*	N	N		
3	Status	*int*	N	N		
4	TopicID	*int*	N	N	**FK**	
5	TeacherID	*int*	N	N	**FK**	
6	Description	*String*	N	N		
7	Price	*money*	N	N		
8	Date	*date*	N	N		
9	Image	*String*	N	N		

4.1.8. Lesson

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*LessonID*	*int*	Y	N	**PK**	Auto-generation id.
2	CourseID	*int*	N	N	**FK**	
3	LessonName	*String*	N	N		
4	Title	*String*	N	N		
5	Document	*String*	N	N		
6	Video	*String*	N	N		
7	Status	*boolean*	N	N		

4.1.9. Order_Detail

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	OrderDetailID	*int*	Y	N	**PK**	Auto-generation id.
2	CourseID	*int*	N	N	**FK**	
3	LessonName	*String*	N	N		
4	Title	*String*	N	N		
5	Document	*String*	N	N		
6	Video	*String*	N	N		
7	Status	*boolean*	N	N		

4.1.10. Order

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	OrderID	*int*	Y	N	**PK**	Auto-generation id.
2	*StudentID*	*int*	N	N	**FK**	
3	Amount	*money*	N	N		
4	Date	*Date*	N	N		

4.1.11. Review Course

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*StaffID*	*int*	Y	N	**PK, FK**	Auto-generation id.
2	*CourseID*	*int*	N	N	**FK, PK**	
3	*Status*	*int*	N	N		
4	*Time*	*Datetime*	N	N		
5	*Comment*	*String*	N	N		

4.1.12. Learn

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*LessionID*	*int*	Y	N	**PK, FK**	Auto-generation id.
2	*StudentID*	*int*	N	N	**FK, PK**	
3	Status	*boolean*	N	N		

4.1.12. User

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*UserID*	*int*	Y	N	**PK**	Auto-generation id.
2	Dob	Date	N	N		
3	UserName	*int*	N	N		
4	*Password*	*String*	N	N		
5	Role	*int*	N	N		
6	Email	*String*	N	N		

4.1.13. Feedback

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*StudentID*	*int*	Y	N	**PK,FK**	Auto-generation id.
2	*CourseID*	*int*	N	N	**PK,FK**	
3	*RatingStar*	*int*	N	N		
4	*Comment*	*String*	N	N		

4.1.14. CartDetail

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*CartID*	*int*	Y	N	**PK,FK**	Auto-generation id.
2	*CourseID*	*int*	N	N	**PK,FK**	

4.1.15. Cart

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*CartID*	*int*	Y	N	**PK**	Auto-generation id.
2	*StudentID*	*int*	Y	N	**PK,FK**	

4.1.16. Teacher_CV

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*CV_ID*	*int*	Y	N	**PK**	Auto-generation id.
2	*TeacherID*	*int*	N	N	**FK**	
3	Date	*Date*	N	N		
4	*StaffID*	*String*	N	N	**FK**	
5	*PDF_Link*	*String*	N	N		
6	*Status*	*int*	N	N		

4.1.17. WithdrawalDetail

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*WithDrawalDetailId*	*int*	Y	N	**PK**	Auto-generation id.
2	*TeacherID*	*int*	N	N	**FK**	
3	*UserID*	*int*	N	N	**FK**	
4	*Money*	*money*	N	N		
5	*Date*	*Date*	N	N		
6	*Status*	*int*	N	N		

4.1.18. Notification

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	*NotificationID*	*int*	Y	N	**PK**	Auto-generation id.
2	*Message*	*string*	N	N		
3	*LinkDirect*	*string*	N	N		

4.1.19. WebcontentImage

#	Field name	Type	Unique	Nullable	PK/ FK	Notes
1	ImageID	*int*	Y	N	**PK**	Auto-generation id.
2	Link	*string*	N	N		

4.1.19. Wallet

Field name	Type	Unique	Nullable	PK/ FK	Notes
WalletID	*int*	Y	N	**PK**	Auto-generation id.
TeacherID	*string*	N	N		
Revenue	*money*	N	N		