

3.4. Class Diagram and Entity class diagram

3.4.1. Overview Class diagram

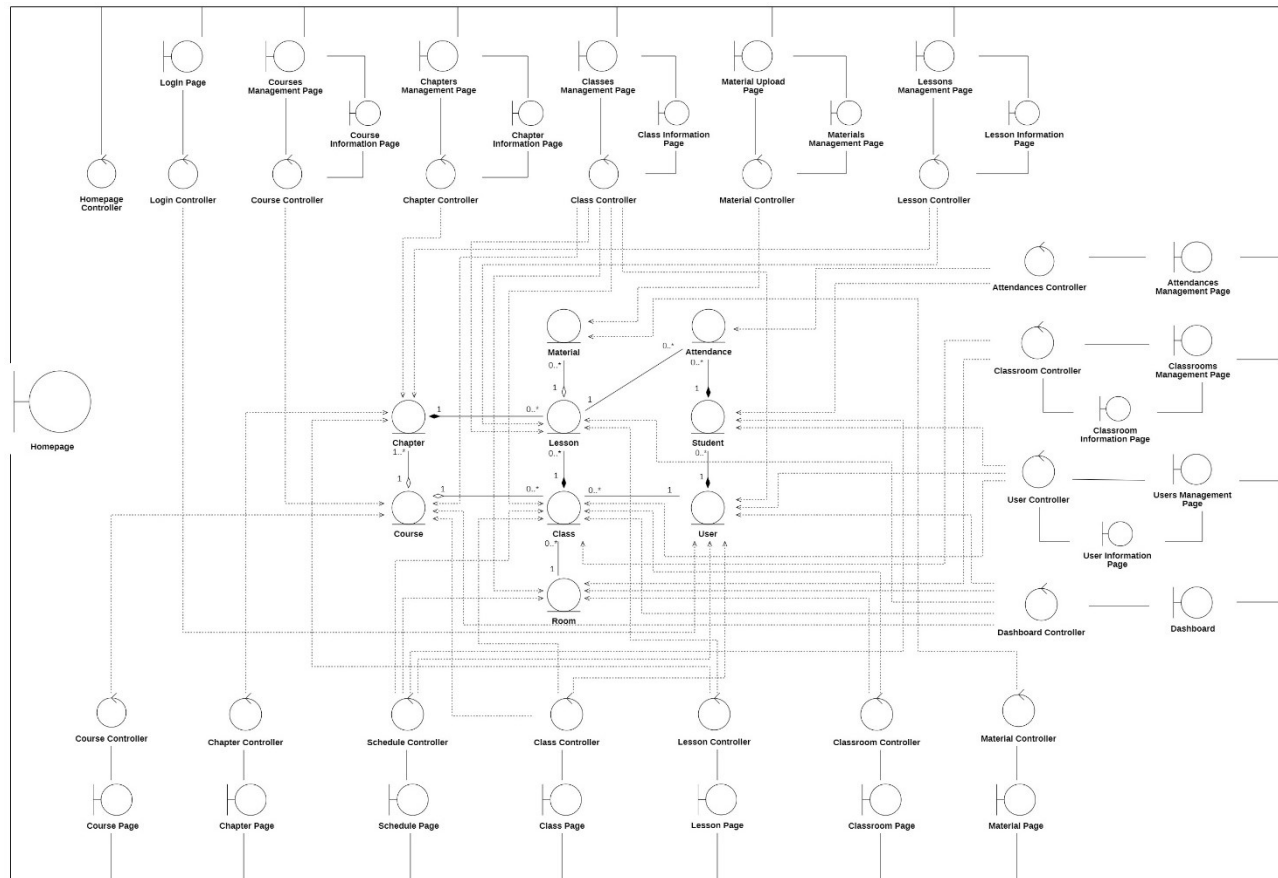


Figure 3. 98. Overview Class Diagram

3.4.2. Entity Class diagram

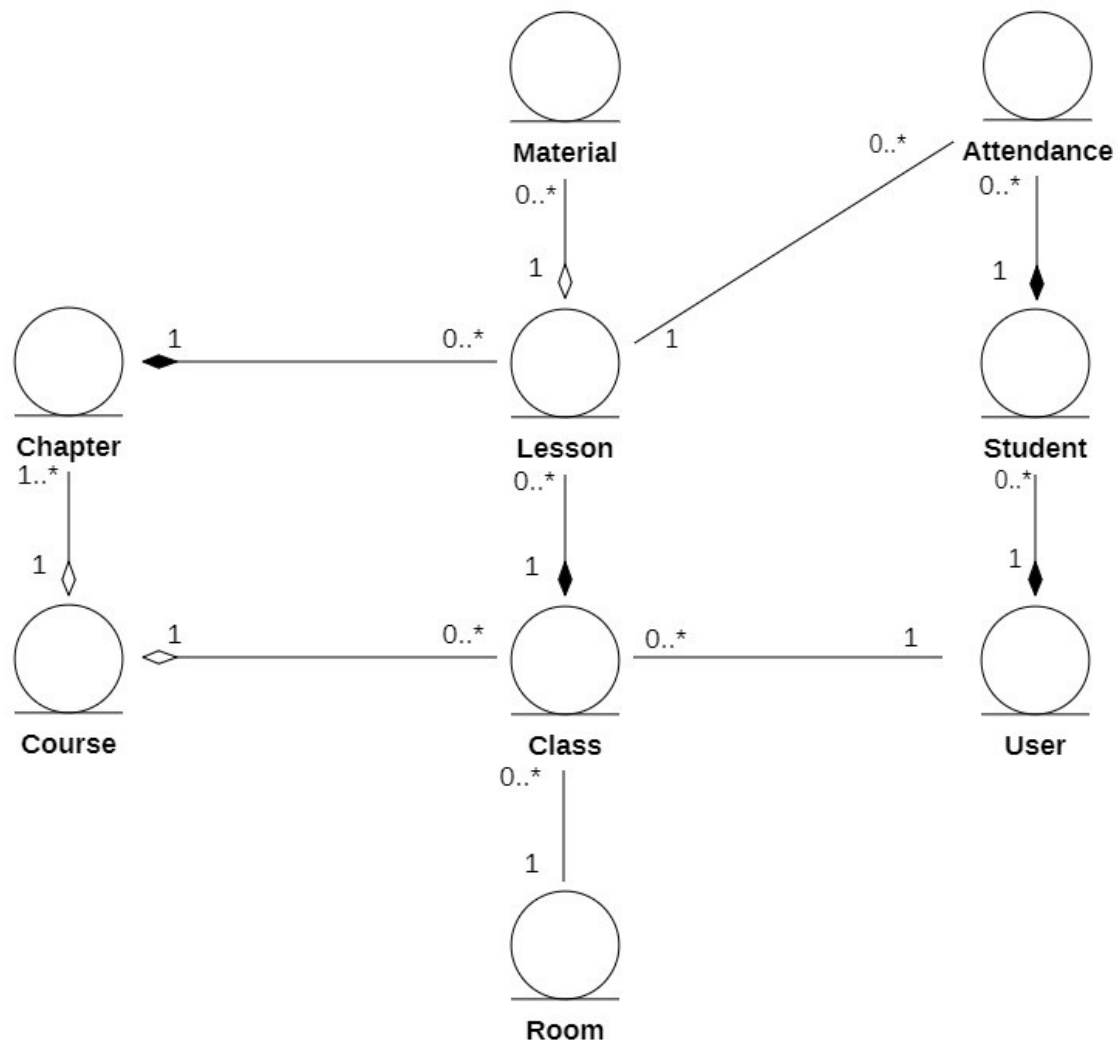


Figure 3. 99. Entity Class Diagram

3.4.3. Class diagram specification

3.4.3.1. Course

Course		
Attribute	Course_ID: String	Course ID generated by system automatically
	Name: String	Name of the Course
	Description: String	Description of the Course
	Course_level: String	Level of the Course base on difficulty
Operation	getNewID(): String	Return new ID for the new Course, unique in database
	QueryData(Conditions): List	Return the result of the query from database base on given conditions
	StoreData(CourseInfo): Boolean	<ul style="list-style-type: none">- Insert a new record based on the new Course information if there is no existing version of that Course yet.- Update the course information if there is already an existing Course in the database. Then return the result of that section (Boolean)

3.4.3.2. Chapter

Chapter		
Attribute	Chapter_ID: String	Chapter ID generated by system automatically
	Name: String	Name of the Chapter
	Description: String	Description of the Chapter
	Category: String	Classify the chapters according to specific skills
	Course_ID: String	The ID of the Course contains this Chapter
Operation	getNewID(): String	Return new ID for the new Chapter, unique in database
	QueryData(Conditions): List	Return the result of the query from database base on given conditions
	StoreData(ChapterInfo): Boolean	<ul style="list-style-type: none">- Insert a new record based on the new Chapter information if there is no existing version of that Chapter yet.- Update the Chapter information if there is already an existing version in the database.

Chapter		
		Then return the result of that section (Boolean)
	DeleteData(ChapterID): Boolean	Delete the Chapter in database base on given ChapterID. Then return the result of that section (Boolean)

3.4.3.3. Class

Class		
Attribute	Class_ID: String	Class ID generated by system automatically
	Name: String	Name of the Class
	DateStart: Date	The Start date of the Class
	DateEnd: Date	The End date of the Class
	SessionTime: Enum	Session Time for schedule of the Class
	SessionDay: Enum	Session Day for schedule of the Class
	Course_ID: String	The ID of the course that this class is currently teaching

Class		
	Teacher_ID: String	The ID of the teacher who is teaching this class.
	Room_ID: String	The ID of the room where this class is currently being held.
Operation	getNewID(): String	Return new ID for the new Class, unique in database
	QueryData(Conditions): List	Return the result of the query from database base on given conditions
	StoreData(ClassInfo): Boolean	<ul style="list-style-type: none"> - Insert a new record based on the new Class information if there is no existing version of that Class yet. - Update the Class information if there is already an existing version in the database. Then return the result of that section (Boolean)
	DeleteData(ClassID): Boolean	Delete the Chapter in database base on given ChapterID. Then return the result of that section (Boolean)

3.4.3.4. Lesson

Lesson		
Attribute	Lesson_ID: String	Lesson_ID generated by system automatically
	LearnDate: Date	The date which the class session is scheduled to teach.
	Class_ID: String	The ID of the class organizing this lesson
	Chapter_ID: String	The ID of the Chapter that this lesson will teach
Operation	getNewID(): String	Return new ID for the new Lesson, unique in database
	QueryData(Conditions): List	Return the result of the query from database base on given conditions
	StoreData(LessonInfo): Boolean	<ul style="list-style-type: none">- Insert a new record based on the new Lesson information if there is no existing version of that Lesson yet.- Update the Lesson information if there is already an existing version in the database. Then return the result of that

Lesson		
		section (Boolean)
	DeleteData(LessonID): Boolean	Delete the Lesson in database base on given LessonID. Then return the result of that section (Boolean)

3.4.3.5. Material

Material		
Attribute	Material_ID: String	Material_ID generated by system automatically
	Path: String	The file path of the material on the computer.
	Lesson_ID: String	The ID of the class that contains this material.
Operation	QueryData(Conditions): List	Return the result of the query from database base on given conditions
	StoreData(MaterialInfo): Boolean	<ul style="list-style-type: none"> - Insert a new record based on the new Material information if there is no existing version of that Material yet. - Update the Material information

Material		
		if there is already an existing version in the database. Then return the result of that section (Boolean)
	DeleteData(MaterialID): Boolean	Delete the Material in database base on given MaterialID. Then return the result of that section (Boolean)

3.4.3.6. Attendance

Attendance		
Attribute	Lesson_ID: String	The ID of the Class in which this attendance is recorded
	Student_ID: String	The ID of the Student which this attendance is recorded
Operation	QueryData(Conditions): List	Return the result of the query from database base on given conditions
	StoreData(AttendanceInfo): Boolean	- Insert a new record based on the new Attendance information if there is no existing version of that Attendance yet. - Update the Attendance

Attendance		
		information if there is already an existing version in the database. Then return the result of that section (Boolean)
	DeleteData(AttendanceID): Boolean	Delete the Attendance in database base on given AttendanceID. Then return the result of that section (Boolean)

3.4.3.7. Room

Room		
Attribute	Room_ID: String	Room_ID generated by system automatically
	Name: String	Name of the classroom
	Capacity: Integer	The maximum capacity of the classroom
Operation	getNewID(): String	Return new ID for the new Classroom, unique in database
	QueryData(Conditions): List	Return the result of the query from database base on given conditions

Room		
	StoreData(classroomInfo): Boolean	<ul style="list-style-type: none"> - Insert a new record based on the new classroom information if there is no existing version of that classroom yet. - Update the classroom information if there is already an existing version in the database. Then return the result of that section (Boolean)
	DeleteData(classroomID): Boolean	Delete the classroom in database base on given classroomID. Then return the result of that section (Boolean)

3.4.3.8. Student

Student		
Attribute	Student_ID: String	Student_ID generated by system automatically
	User_ID: String	The ID of the student as a user in system
	Class_ID: Integer	The ID of the class that this student is currently enroll

Student		
Operation	getNewID(): String	Return new ID for the new Student, unique in database
	QueryData(Conditions): List	Return the result of the query from database base on given conditions
	StoreData(StudentInfo): Boolean	<p>- Insert a new record based on the new student information if there is no existing version of that student yet.</p> <p>- Update the student information if there is already an existing version in the database.</p> <p>Then return the result of that section (Boolean)</p>
	DeleteData(StudentID): Boolean	Delete the student in database base on given studentID. Then return the result of that section (Boolean)

3.4.3.9. User

User		
Attribute	User_ID: String	Student_ID generated by system automatically
	Username: String	The account name of user use for login into system
	Password: Integer	The password of user use for login into system
	Role: String	The role of user in system
	DisplayName: String	The name of user that display in system interface
Operation	getNewID(): String	Return new ID for the new User, unique in database
	QueryData(Conditions): List	Return the result of the query from database base on given conditions
	StoreData(UserInfo): Boolean	<ul style="list-style-type: none">- Insert a new record based on the new User information if there is no existing version of that User yet.- Update the User information if there is already an existing

User		
		version in the database. Then return the result of that section (Boolean)
	DeleteData(UserID): Boolean	Delete the User in database base on given UserID. Then return the result of that section (Boolean)
	ValidateUserCredentialinDB(UserCredentials): Boolean	Validate user credential in database for login action

3.4.3.10. Login Controller

Login Controller		
Operation	ValidateUserCredentials(UserCredential): Boolean	Request Database to check User credentials then return result of the validation

3.4.3.11. Course Controller

Course Controller		
Operation	GetListOfCourses(): List	Request Database to get list of all existed Courses
	GetCourseInfo(CourseID): Course	Request database to get Course information base on given CourseID
	AddCourse(CourseInfo): Boolean	Request database to insert new Course with given Course information. Return validation result
	UpdateCourse(CourseInfo): Boolean	Request database to update Course with given Course information. Return validation result
	DeleteCourse(CourseID): Boolean	Request database to delete Course base on given CourseID. Return validation result
	SearchCourse(Keyword): List	Request database to get list of Courses related to the given Keyword

3.4.3.12. Chapter Controller

Chapter Controller		
Operation	GetListOfChapters(): List	Request Database to get list of all existed Chapters
	GetChapterInfo(ChapterID): Chapter	Request database to get Chapter information base on given ChapterID
	AddChapter(ChapterInfo): Boolean	Request database to insert new Chapter with given Chapter information. Return validation result
	UpdateChapter(ChapterInfo): Boolean	Request database to update Chapter with given Chapter information. Return validation result
	DeleteChapter(ChapterID): Boolean	Request database to delete Chapter base on given ChapterID. Return validation result
	SearchChapter(Keyword): List	Request database to get list of Chapters related to the given Keyword

3.4.3.13. Class Controller

Class Controller		
Operation	GetListOfClasses(): List	Request Database to get list of all existed Classes
	GetClassInfo(ChapterID): Class	Request database to get Class information base on given ClassID
	AddClass(ClassInfo): Boolean	Request database to insert new Class with given Class information. Return validation result
	UpdateClass(ClassInfo): Boolean	Request database to update Class with given Class information. Return validation result
	DeleteClass(ClassID): Boolean	Request database to delete Class base on given ClassID. Return validation result
	SearchClass(Keyword): List	Request database to get list of Classes related to the given Keyword
	GenerateReport(ClassInformation): Report	Request database to query data for report then generate report for specific class

Class Controller		
	ExportReport(Report): Boolean	Export pdf version of report then process save file to user's computer

3.4.3.14. Lesson Controller

Lesson Controller		
Operation	GetListOfLessons(): List	Request Database to get list of all existed Lessons
	GetLessonInfo(LessonID): Class	Request database to get Lesson information base on given LessonID
	AddLesson(LessonInfo): Boolean	Request database to insert new Lesson with given Lesson information. Return validation result
	UpdateLesson(LessonInfo): Boolean	Request database to update Lesson with given Lesson information. Return validation result
	DeleteLesson(LessonID): Boolean	Request database to delete Lesson base on given LessonID. Return validation result

Lesson Controller		
	SearchLesson(Keyword): List	Request database to get list of Lessons related to the given Keyword

3.4.3.15. Material Controller

Material Controller		
Operation	GetListOfMaterials(): List	Request Database to get list of all existed Materials
	GetMaterial(MaterialID): Material	Request database to get Material information base on given MaterialID
	AddMaterial(MaterialInfo): Boolean	Request database to insert new Material with given Material information. Return validation result
	DownloadMaterial(MaterialInfo): Boolean	Base on Material information, process download Material for user
	DeleteMaterial(MaterialID): Boolean	Request database to delete Material base on given MaterialID. Return validation result

Material Controller		
	SearchMaterial(Keyword): List	Request database to get list of Materials related to the given Keyword
	ValidateFile(UploadFile): Boolean	Validate file upload from user, return result of validation

3.4.3.16. Attendance Controller

Attendance Controller		
Operation	GetListOfAttendances(): List	Request Database to get list of all existed Attendances
	GetAttendanceInfo(AttendanceID) : Class	Request database to get Attendance information base on given AttendanceID
	AddAttendance(AttendanceInfo): Boolean	Request database to insert new Attendance with given Attendance information. Return validation result
	UpdateAttendance(AttendanceInfo): Boolean	Request database to update Attendance with given Attendance information. Return validation result

Attendance Controller		
	DeleteAttendance(AttendanceID): Boolean	Request database to delete Attendance base on given AttendanceID. Return validation result
	SearchAttendance(Keyword): List	Request database to get list of Attendances related to the given Keyword

3.4.3.17. Classroom Controller

Classroom Controller		
Operation	GetListOfClassrooms(): List	Request Database to get list of all existed Classrooms
	GetClassroomInfo(MaterialID): Class	Request database to get Classroom information base on given ClassroomID
	AddClassroom(ClassroomInfo): Boolean	Request database to insert new Classroom with given Classroom information. Return validation result
	UpdateClassroom(ClassroomInfo): Boolean	Request database to update Classroom with given Classroom information. Return

Classroom Controller		
		validation result
	DeleteClassroom(ClassroomID): Boolean	Request database to delete Classroom base on given ClassroomID. Return validation result
	SearchClassroom(Keyword): List	Request database to get list of Classrooms related to the given Keyword
	GenerateSchedule(ClassroomID)	Request data from database base on ClassroomID then generate Schedule for this room

3.4.3.18. User Controller

User Controller		
Operation	GetListOfUsers(): List	Request Database to get list of all existed Users
	GetUserInfo(UserID): Class	Request database to get User information base on given UserID
	AddUser(UserInfo): Boolean	Request database to insert new User with given User

User Controller		
		information. Return validation result
	UpdateUser(UserInfo): Boolean	Request database to update User with given User information. Return validation result
	DeleteUser(UserID): Boolean	Request database to delete User base on given UserID. Return validation result
	SearchUser(Keyword): List	Request database to get list of Users related to the given Keyword
	EnrollStudent(UserID, ClassID)	Process Class Enrollment for selected user

3.4.3.19. Dashboard Controller

Dashboard Controller		
Operation	QueryData(AnalyzeCategory): List	Request Database to prepare data for selected category
	GenerateChart(PreparedData): Chart	Generate Chart for prepared data then return a visual Chart

3.4.3.20. Schedule Controller

Schedule Controller		
Operation	QueryData(): List	Request Database to prepare data for schedule
	GenerateSchedule(UserID): Schedule	Generate schedule for User with data queried from database

3.4.3.21. Homepage Controller

Schedule Controller		
Operation	ProcessLogout(): Void	Process Logout request for user