MINISTRY OF
EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

Chess Online Learning System

GROUP 04	
Group members	Phạm Hoàng Tuyết Ngân – SE62778 (Leader) Lưu Duy Hòa – SE62529 Đặng Văn Hoàng – SE62687
Supervisor	Nguyễn Huy Hùng
Ext. Supervisor	N/A
Capstone Project code	COLS

- Ho Chi Minh City, ***13th May, 2019*** -

This page is intentionally left blank

Table of Contents

List of Table **6**
List of Figure **9**
A. Introduction **13**
1. Project Information 13
2. Introduction 13
3. Current Situation 13
4. Problem Definition 14
5. Proposed Solution 14
5.1 Feature functions 14
5.2 Benefits and Trade-offs 14
6. Functional Requirements 14
7. Role and Responsibility 16
B. Software Project Management Plan **16**
1. Problem Definition 16
1.1 Name of this Capstone Project 16
1.2 Problem Abstract 16
1.3 Project Overview 16
2. Project organization 20
2.1 Software Process Model 20
2.2 Roles and responsibilities 20
2.3 Tools and Techniques 21
3. Project Management Plan 22
3.1 Product Backlog 22
3.2 Sprint Backlog 22
3.3 Meeting Minutes 22
4. Coding Convention 22
C. Software Requirement Specification **23**
1. User Requirement Specification 23
1.1 Guest Requirement 23
1.2 Authorized user Requirement 23
1.3 Learner Requirement 23
1.4 Instructor Requirement 23
1.5 Admin Requirement 24
1.6 System Bot Requirement 24
2. System Requirement Specification 24

2.1 External Interface Requirement 24
2.2 System Overview Use case[1] 25
2.3 List of Use Case 27
3. Software System Attribute 91
3.1 Usability 91
3.2 Reliability 91
3.3 Availability 91
3.4 Security 91
3.5 Maintainability 91
3.6 Portability 91
3.7 Performance 91
4. Conceptual diagram 92
D. Software Design Description 93
1. Design Overview 93
2. System Architectural Design 94
2.1 System Architecture Overview[2] 94
3. Component Diagram[1] 97
4. Detail Description 98
4.1 Class Diagram[1] 98
4.2 Class Diagram Explanation 101
4.3 State Machines[1] 107
4.4 Interactive Diagram[1] 108
5. User Interface Design 119
5.1 Learner Interface Design 119
5.2 Instructor Interface Design 135
5.3 Admin Interface Design 145
6. Database Design 148
6.1 Entity relationship diagram (ERD)[3] 148
6.2 Entity dictionary 149
7. Algorithms 149
7.1 Convert chess move list from database to display list 149
7.2 Building chess bot 152
7.3 Building Elo rating system 156
7.4 Finding Similar User 157
7.5 Course Rating Prediction (User-based filtering) 158
7.6 Item-based filtering 158

E. System Implementation & Test **159**
1. Introduction 159
1.1 Overview 159
1.2 Test approach 160
2. Database Relationship Diagram 160
2.1 Physical diagram 160
2.2 Data dictionary 162
3. Test plan 166
3.1 Features to be tested 166
3.2 Features not to be tested 166
4. System Testing Test Case 166
4.1 Core Flow 166
4.2 Test case 168
F. Software User's Manual **184**
1. Installation Guide 184
1.1 Setting up environment at server side 184
1.2 Deployment 184
2. User guide 196
2.1 Guest 196
2.2 Learner 198
2.3 Instructor 203
2.4 Admin 209
G. Appendix **213**

List of Table

Table 1: Roles and Responsibilities....16
Table 2: Hardware Requirement for Server....19
Table 3: Hardware Requirement for PC....19
Table 4: Software requirement....19
Table 5: Roles and Responsibilities Details....21
Table 6: Tools....21
Table 7: Techniques....21
Table 8: Sign In specification....28
Table 9: Sign Up specification....30
Table 10: View Published Courses specification....32
Table 11: View course details specification....33
Table 12: Search Courses specification....34
Table 13: Filter Courses specification....35
Table 14: Enrol Course specification....37
Table 15: View Enrolled Course specification....38
Table 16: : Create Review specification....39
Table 17: Edit Review specification....41
Table 18: Remove Review specification....42
Table 19: Read Interactive Lesson specification....44
Table 20: Read Uninteractive Lesson specification....45
Table 21: Do Exercise specification....47
Table 22: View Learning Progress specification....48
Table 23: View List Of Suggested Course specification....50
Table 24: Play Chess With The Bot specification....51
Table 25: Play Chess With Learner specification....53
Table 26: Update Learner Skill Level specification....54
Table 27: Make Course Suggestion For Learner specification....55
Table 28: Create Course specification....57
Table 29: Edit Course specification....59
Table 30: Edit Course Status specification....60
Table 31: View Course Reports specification....61
Table 32: Create Interactive Lesson specification....64
Table 33: Edit Interactive Lesson specification....66
Table 34: Create Uniteractive Lesson specification....68
Table 35: <Instructor> Edit Uninteractive Lesson sepecification....69
Table 36: <Instructor> Create Exercise specification....72
Table 37: Edit Exercise specification....75
Table 38: Remove Lesson (Exercise) specification....76
Table 39: Create Category specification....78
Table 40: Edit Category specification....79
Table 41: Remove Category specification....80
Table 42: View User Reports specification....81
Table 43: View Bot Reports specification....82
Table 44: View Course Reports specification....83
Table 45: Review New Course specification....85
Table 46: View Course List specification....86
Table 47: Unpublish Course specification....87

Table 48: Review Instructor specification....89
Table 49: View User List specification....89
Table 50: Active/Deactivate User specification....91
Table 51: Conceptual diagram data dictionary....93
Table 52: Front-end Servers Component Dictionary....97
Table 53: Back-end Servers Component Dictionary....98
Table 54: Class Dictionary....101
Table 55:User attributes....101
Table 56: Role attributes....101
Table 57: Notification attributes....102
Table 58: Certificate attributes....102
Table 59: PointLog attributes....102
Table 60: GameHistory attributes....103
Table 61: GameHistoryStatus attributes....103
Table 62: Course attributes....104
Table 63: CourseStatus attributes....104
Table 64: CourseLesson attributes....104
Table 65: Review attributes....105
Table 66: UserHasCourse attributes....105
Table 67: UserHasCourseStatus attributes....105
Table 68: Category discussion....105
Table 69: CourseCategory attributes....106
Table 70: Lesson attributes....106
Table 71: LessonType attributes....106
Table 72: LearningLog attributes....107
Table 73: Home Page –Buttons/Hyperlinks....119
Table 74: View Published Course –Fields and Buttons/Hyperlinks....121
Table 75: Before Enrol – Fields and Buttons/Hyperlinks....122
Table 76: After Enrol – Buttons/Hyperlinks....122
Table 77: Review Course – Fields and Buttons/Hyperlinks....124
Table 78: Lesson List – Fields and Buttons/Hyperlinks....125
Table 79: Learn Interactive Lesson – Fields and Buttons/Hyperlinks....126
Table 80: Move Container – Fields and Buttons/Hyperlinks....127
Table 81: Learn Uninteractive Lesson – Fields and Buttons/Hyperlinks....128
Table 82: Do Exercise – Fields and Buttons/Hyperlinks....129
Table 83: Finish Exercise – Fields and Buttons/Hyperlinks....129
Table 84: Play Chess – Buttons/Hyperlinks....130
Table 85: Game Control – Fields and Buttons/Hyperlinks....131
Table 86: Playing - Fields....132
Table 87: Account Information – Fields and Buttons/Hyperlinks....133
Table 88: Enrolled Course - Fields....134
Table 89: Instructor Menu –Buttons/Hyperlinks....135
Table 90: Create course – Fields and Buttons/Hyperlinks....136
Table 91: Course List – Fields and Buttons/Hyperlinks....138
Table 92: Course Overview – Fields and Buttons/Hyperlinks....139
Table 93: Lesson List - Fields....139
Table 94: Setiings – Buttons/Hyperlinks....140
Table 95: Add Interactive Lesson – Fields and Buttons/Hyperlinks....141
Table 96: Add Uninteractive Lesson – Fields and Buttons/Hyperlinks....142

Table 97: Add Exercise – Fields and Buttons/Hyperlinks 143
Table 98: Dynamic Chess position – Fields and Buttons/Hyperlinks 144
Table 99: Preview Lesson – Fields and Buttons/Hyperlinks 145
Table 100: Create Category –Fields and Buttons/Hyperlinks 145
Table 101: List of New Instructor – Fields and Buttons/Hyperlinks 146
Table 102: Instructor Details - Fields 147
Table 103: Entity data dictionary 149
Table 104: Move list dictionary 151
Table 105: Data Table Dictionary 162
Table 106: Data Table Description 166
Table 107: Test cases <Guest> 171
Table 108: Test cases <Learner> 176
Table 109: Test cases <Instructor> 181
Table 110: Test cases <Admin> 183
Table 111: Hardware Requirement for Server 184
Table 112: Software Requirement for Server. 184
Table 113: Search Course description 197
Table 114: View Course Details description 197
Table 115: Enrol Course description 198
Table 116: Learn Lesson description 199
Table 117: Create Review description 200
Table 118: Manage Review desciprtion 201
Table 119: Update Profile description 202
Table 120: Manage Enrolled Courses description 202
Table 121: Create Course description 203
Table 122: Edit Course Status description 204
Table 123: Manage Lesson description 205
Table 124: Create Interactive Lesson description 206
Table 125: Create Uninteractive Lesson description 207
Table 126: Create Exercise description 208
Table 127: Active/Deactive Account description 209
Table 128: Search Account description 210
Table 129: Accept/Reject Instructor Registration description 210
Table 130: Manage Category description 211
Table 131: Manage Notification description 212

List of Figure

Figure 1: Scrum Model 20
Figure 2: System Overview Use Cas 26
Figure 3: <Guest> Overview Use Case 27
Figure 4: <Guest> Sign In 27
Figure 5: <Guest> Sign Up 28
Figure 6: <Guest & Learner> View Published Courses 31
Figure 7: <Guest & Learner> View Course Details 32
Figure 8: <Guest & Learner> Search Courses 33
Figure 9: <Guest & Learner> Filter Courses 34
Figure 10: <Learner> Overview Use Case 35
Figure 11: <Learner> Enrol Course 36
Figure 12: <Learner> View Enrolled Course 37
Figure 13: <Learner> Create Review 38
Figure 14: <Learner> Edit Review 40
Figure 15: <Learner> Remove Review 41
Figure 16: <Learner> Read Interactive Lesson 42
Figure 17: <Learner> Read Uninteractive Lesson 44
Figure 18: <Learner> Do Exercise 46
Figure 19: <Learner> View Learning Progress 48
Figure 20: <Learner> View List Of Suggested Course 49
Figure 21: <Learner> Play Chess With The Bot 50
Figure 22: <System Bot> Overview Use Case 52
Figure 23: <System Bot> Play Chess With Learner 52
Figure 24: <System Bot> Update Learner Skill Level 53
Figure 25: <System Bot> Make Course Suggestion For Learner 54
Figure 26: <Instructor> Overview Use Case 55
Figure 27: <Instructor> Create Course 55
Figure 28: <Instructor> Edit course 57
Figure 29: <Instructor> Edit Course Status 59
Figure 30: <Instructor> View Course Reports 60
Figure 31: <Instructor> Create Interactive Lesson 62
Figure 32: <Instructor> Edit Interactive Lesson 64
Figure 33: <Instructor> Create Uninteractive Lesson 66
Figure 34: <Instructor> Edit Uninteractive Lesson 68
Figure 35: <Instructor> Create Exercise 70
Figure 36: <Instructor> Edit Exercise 72
Figure 37: <Instructor> Remove Lesson (Exercise) 75
Figure 38: <Admin> Overview Use Case 76
Figure 39: <Admin> Create Category 76
Figure 40: <Admin> Edit Category 78
Figure 41: <Admin> Remove Category 79
Figure 42: <Admin> View User Reports 81
Figure 43: <Admin> View Bot Reports 81
Figure 44: <Admin> View Course Reports 82
Figure 45: <Admin> Review New Course 83
Figure 46: <Admin> View Course List 85
Figure 47: <Admin> Unpublish Course 86

Figure 48: <Admin> Review Instructor 87
Figure 49: <Admin> View User List 89
Figure 50: <Admin> Active/Deactivate User 90
Figure 51: Conceptual diagram 92
Figure 52: System architecture overview 95
Figure 53: Front-end Servers Component diagram 97
Figure 54: Back-end Server Component diagram 98
Figure 55: Class Diagram 99
Figure 56: Course State 107
Figure 57: Learner Has Course State 108
Figure 58: Sequence Diagram - <Guest & Learner> View Published Course 108
Figure 59: Sequence Diagram - <Guest & Learner> Search Courses 109
Figure 60: Sequence Diagram - <Guest & Learner> View Course Details 109
Figure 61: Sequence Diagram - <Learner> Enrol Course 110
Figure 62: Sequence Diagram - <Learner> Create Review 111
Figure 63: Sequence Diagram - <Learner> View List Of Suggested Course 111
Figure 64: Sequence Diagram - <Instructor> Create Course 112
Figure 65: Sequence Diagram - <Instructor> Submit Course 113
Figure 66: Sequence Diagram - <Instructor> Create Lesson 114
Figure 67: Sequence Diagram - <Instructor> Edit Lesson 115
Figure 68: Sequence Diagram - <Instructor> Remove Lesson 116
Figure 69: Sequence Diagram - <Admin> Update Course Status 117
Figure 70: Sequence Diagram - <Admin> Update Course Status 118
Figure 71: Home Page 119
Figure 72: View Published Course 120
Figure 73: Before Enrol 121
Figure 74: After Enrol 122
Figure 75: Review Course 123
Figure 76: Lesson List 124
Figure 77: Learn Interactive Lesson 125
Figure 78: Move Container 126
Figure 79: Learn Uninteractive Lesson 127
Figure 80: Do Exercise 128
Figure 81: Finish Exercise 129
Figure 82: Play Chess 130
Figure 83: Game control 131
Figure 84: Playing 132
Figure 85: Play Chess 133
Figure 86: Game control 134
Figure 87: Instructor Menu 135
Figure 88: Create course 136
Figure 89: Course List 137
Figure 90: Course Overview 138
Figure 91: Lesson List 139
Figure 92: Settings 140
Figure 93: Add Interactive Lesson 140
Figure 94: Add Uninteractive lesson 141
Figure 95: Add Exercise 142
Figure 96: Dynamic Chess Position 143

Figure 97: Preview Lesson 144
Figure 98: Create Category 145
Figure 99: List of New Instructor 146
Figure 100: Instructor Details 147
Figure 101: Entity relationship diagram (ERD) 148
Figure 102: Target Move List 150
Figure 103: Sample move list from database 150
Figure 104: Example simple linked list 151
Figure 105: Example tree 151
Figure 106: Move list result 152
Figure 107: A chessboard with square in AN 153
Figure 108: Sample rank in FEN String 153
Figure 109: Sample Position in FEN String 154
Figure 110: Building Chess Bot Flow chart 155
Figure 111: Physical diagram 161
Figure 112: <Guest> Core Flow 166
Figure 113: <Learner> Core Flow 167
Figure 114: <Instructor> Core Flow 167
Figure 115: <Admin> Core Flow 167
Figure 116: <MySQL> Select Service 186
Figure 117: <MySQL> Create Database 186
Figure 118: <MySQL> Select Database 187
Figure 119: Input Information 187
Figure 120: <MySQL> Configuration 188
Figure 121: <MySQL> Edit Rule 188
Figure 122: <MySQL> Using in Local 189
Figure 123: COLS Database 190
Figure 124: <Web Server> Select Service 190
Figure 125: <Web Server> Start Elastic Beanstalk 191
Figure 126: <Web Server> Input Information 191
Figure 127: <Web Server> Manage Web Service 192
Figure 128: <Web Server> Upload File 192
Figure 129: <Web Server> Result 192
Figure 130: <Front-end App> Select Service 193
Figure 131: <Front-end App> Create Bucket 193
Figure 132: <Front-end App> Input Information 194
Figure 133: <Front-end> App Navigate To a bucket 194
Figure 134: <Front-end App> Select Upload 195
Figure 135: <Front-end App> Input Information and Click On Upload Button 195
Figure 136: <Front-end App> Configuration 196
Figure 137: Search Course 196
Figure 138: View Course Details 197
Figure 139: Enrol Course 198
Figure 140: Learn Lesson 199
Figure 141: Create review 200
Figure 142: Manage Review 201
Figure 143: Update profile 201
Figure 144: Manage course enrolled 202
Figure 145: Create course 203

Figure 146: Edit Course Status 204
Figure 147: Manage Lesson 205
Figure 148: Create Interactive Lesson 205
Figure 149: Create Uninteractive Lesson 207
Figure 150: Create Exercise 207
Figure 151: Active/Deactive Account 209
Figure 152: Search User 209
Figure 153: Accept/Reject Instructor Registration 210
Figure 154: Manage Category 211
Figure 155: Manage Notification 212

A. Introduction

1. Project Information

- Project name: **Chess Online Learning System**
- Project Code: **COLS**
- Product Type: **Website application**
- Start Date: **13th May, 2019**
- End Date: **October, 2019**

2. Introduction

In this document, we introduce an e-learning portal for people who are willing to learn, improve and entertain the chess. Chess is a strategic board game for 2 players. Chess helps people to relax and improve their thinking. Nowadays, chess is a popular sport in the world. Currently, there are so many ways, which learn chess such as reading books, directly teaching and watching the online course. In a short research from wikihow.com, the most effective learning way is direct teaching or face to face practice, but it also takes too much cost. Meanwhile, learning online from e-portal currently, are the most successful way. However, there are some preventions in learning on the e-portal which could be named like confliction in communicating, boring traditional learning, non-evaluating ranked leaner and non-suggestion road for better playing. At last, reading book is the worst effective learning chess for general people, because it does not interact directly with the learner. Based on our researches and analysis, we proposed an e-learning system to help people to learn chess and improve their level. Especially, the main target is Vietnamese people.

In our system, the player can learn chess by tutorial and practice with exercises, and the instructor can post their course to system. Moreover, the system can be a bridge between chess players and instructors by providing a platform to help their communication.

This document also describes our working process in four months includes our perspective in the system, component designs, and detailed core workflows. We hope the system and our solution will help to resolve the problems from learning chess for the Vietnamese community.

3. Current Situation

When starting with chess, generally people are going to search for information and looking for a suitable tutorial for themselves online or offline. In the offline method, both instructors and learners are depended on their time, location and costly for infrastructure. By the way, for the online method, the most factor that brings the pain to leaner that is going to make them avoid choosing a website is language. Not only for a learner who does not know the common language, but also for the learner who knows. The second problem is the tutorials, which is not highly detailed or has bored traditional teaching ways. Therefore, the tutorial is static while the learner always wants to be challenged. Moreover, it causes frustration inside the learner.

4. Problem Definition

From the above current situation, we recognize some disadvantages:

- There is some online interactive courses chess (chess.com/lessons, lichess.org/learn#), but it is fully English, so Vietnamese people are hard to learn.
- Some chess course by Vietnamese are provided, but they do not have any place to practice.
- An instructor, who wants to teach chess must prepare a place with full infrastructure.
- An offline learner must depend on instructor about learning time and location.

5. Proposed Solution

Our proposed solution is building a system, which named is Chess Online Learning System (COLS) for Vietnamese people with the following functions:

5.1 Feature functions

- Provide interactive chess courses with practice environment by Vietnamese for the learner.
- Provide an environment for the instructor, who wants to share their chess knowledge.
- Provide chess bot, who can play automatically with the learner.
- Collect learner learning history and suggest suitable courses.
- Support Elo rating system to classify the level of learners and courses. A learner can only learn courses which are equal or lower than their level.

5.2 Benefits and Trade-offs

- Benefits:
  - An instructor does not need to spend the cost to prepare face to face classes.
  - Flexible in learning time and place.
  - Everything is free.
  - Connecting people who want to learn chess and share chess knowledge without caring about their location.
  - A learner can practice conveniently in our system.
- Trade-offs:
  - The type of lesson or exercise is limited.
  - The course and user activities are checking manually.
  - The instructor benefits are unclear, there is no method to collect money for them, and it will be developed in the next phase.
  - There is only support to play chess with an automated bot, without supporting peer-to-peer playing.

6. Functional Requirements

Function requirements of the system are listed below:

- Authentication and Authorization:

    - Authenticate and authorize users to learn or manage system content and user.
- New account registration:
    - A guest can sign up to be a learner or an instructor.
    - A new learner must estimate their chess skill level to start the learning progress. Skill level will be converted to a point for the learner.
    - A new instructor must apply their information to prove competence.
- Manage profile
    - An authorized user can view and edit his/her information.
- View courses
    - A guest or learner can search for any course by course name.
    - A guest or learner can filter courses by category and skill level.
    - A guest or learner can view course details.
- Enrol course
    - A learner can enrol their wanted course if they have enough skill level.
- Learn course
    - A learner can learn any course which he/she have enrolled.
    - A learner can review course he/she has already learned.
    - Learning includes learn chess game, static content and do exercises
    - A learner can manage his/her learning progress.
- Course suggestion
    - A learner can view suggested list which he/she should learn.
- Manage course
    - An instructor can manage his/her courses such as lessons, exercises or feedbacks.
    - An instructor can submit his/her course and wait for review. If a course is rejected, the instructor can restore them else if a course is approved to publish, the instructor can unpublish their course to re-build content.
    - An admin can publish or unpublish the course.
- Manage category
    - A course must be classified to some categories, which were given from system.
    - An admin can manage system categories.
- Manage user
    - An admin can control learner or instructor account status.
    - An admin can review new instructor registration.
- Play chess
    - A learner can play chess with a bot.
    - Learners can increase their point by win or draw games.
    - Learners can lose their point if they lose the game.
- Manage the learner skill level
    - An Elo rating system is applied to calculate learner point and evaluate learner chess skill level. Each level will be linked with point range.
    - The system can classify learner skill level based on their point.

7. Role and Responsibility

No	Full Name	Role	Position	Contact
1	Nguyễn Huy Hùng	Project Manager	Supervisor	hungnh@fpt.edu.vn
2	Phạm Hoàng Tuyết Ngân	Developer	Leader	nganphtse62778@fpt.edu.vn
3	Lưu Duy Hòa	Developer	Member	hoaldse62529@fpt.edu.vn
4	Đặng Văn Hoàng	Developer	Member	hoangdvse62687@fpt.edu.vn

Table 1: Roles and Responsibilities

B. Software Project Management Plan

1. Problem Definition

1.1 Name of this Capstone Project

- Official name: Chess Online Learning System
- Vietnamese name: Hệ thống dạy cờ vua
- Abbreviation: COLS

1.2 Problem Abstract

Learning is a combination of reading and practice, reading for getting knowledge and practising for confirming your knowledge. It also right for chess learning. Currently, some sites support chess playing guide for Vietnamese, but they only contain static content with a boring user interface.

There are some popular chess sites support learning chess as known as: chess.com, lichess,... but their content is fully in English, then it is a barrier for Vietnamese.

In this project, we build a system call Chess Online Learning System (COLS) to support interactive learning like chess.com, lichess for Vietnamese with included features: playing chess, learning chess by chess game, text resource and solve chess problems.

We also provide a platform for instructors, who have more experiences with chess to sharing their knowledge.

1.3 Project Overview

1.3.1 Current Situation

There are project situations which were analysed following SWOT model:

<u>Strengths:</u>

- **Software development knowledge:** All team members have basic knowledge about software development, especially is web application development.
- **Willing to learn new things:** All team members were learned basic knowledge about computer programming, then we are willing to learn new things to apply in project.
- **Chess knowledge:** All members also have a basic knowledge of chess.

<u>Weakness:</u>

- **Chess programming:** There no members have experience in chess programming, such as handle chessboard, chess move data in the program.
- **Chess learning content:** Our team does not have enough knowledge of building chess learning content.
- **Technology:** All of the team members do not have experience in working with frameworks.

Opportunities:

- **Chess libraries:** There are some open-source libraries about chess programming and all of it is relatively stable to use
- **Chess document:** There are many chess books which were published, we will use the series of Mr. Luong Trong Minh for demonstration.

Threats:

- **Meeting time:** Each team member have different free time, then we do not have the more same time for offline meeting.

1.3.2 The Proposed System

- The system has a learning portal which supports chess learning and practice features.
- The system also has an admin portal which supports course management of instructors and administrators. Admin also uses this portal to manage the system user account.
- Currently, we use Google for authentication and we also manage user information to perform authorization.
- Database and server-app will be hosted in the cloud to decrease coding integration time and it can work 24/7.
- Chess rendering libraries are used in client and a web-socket will be created to synchronize chess game playing status.
- The system will have three sub-systems:
  + A learning application, which supports the learner to learn and practice.
  + A admin application for instructor and admin to build system content and manage the system.
  + A server application to access database, perform system business logic with data and serve API for front-end applications.

1.3.2.1 Learning Application

- For guest:
  - View courses and course details in the system
  - Search course by name
  - Filter course by category and skill level
  - Sign up as a learner or instructor and sign in by Google mail
- For authenticated user:
  - View and update profile
- For learner:
  - View courses and search or filter courses like a guest
  - View list of suggested courses
  - Enrol course with required chess skill level

    - Finish chess course by reading all lessons and pass all exercises
    - Review learned course
    - View learning processing
    - Play chess

1.3.2.2 Admin Application

- For authenticated user:
    - View and update profile
- For instructor:
    - Add or edit course
    - Publish or unpublish course
    - Build course content: lessons and exercises
    - View statistics about their course
- For admin:
    - Change user accounts status
    - Review course
    - Review instructor
    - Publish waiting course or unpublish published course
    - View statistics about system courses and users
    - Create, edit or remove category

1.3.2.3 Server Application

- Access database and perform system business logic
- Perform scheduled tasks
- Provide API for front-end applications

1.3.3 Boundaries of the System

This is features list which current system supports:

- The system only supports for the web browser.
- The system only supports playing chess player to the bot, not player to player.
- The lesson content has 2 types: static content and chess game analyse content.
- The exercise answer has 2 types: automatic answer and manual answer.
- System Elo rating system is based on common Elo rating and learner can only improve chess skill level by win or draw chess game to increase their point.
- The system only suggests course based on user learning history and similar user in the database.

1.3.4 Future Plan

Current system also has problems and it will be developed in the future:

- Detect cheating in game to improve point reliability.
- Support chess playing peer to peer to extend practice opportunity for learner.
- Develop money collecting features for instructor.
- Automatively administrator working.
- Develop course suggestion feature based on learner profile.

1.3.5 Development Environment

1.3.5.1 Hardware requirements

Server	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (8 Mbps)	Cable, Wi-Fi (50 Mbps)
Operating System	Window Server 2008	Window Server 2012
Computer Processor	Intel® Xeon ® 1.4GHz	Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz)
Computer Memory	4GB RAM	32GB RAM or more
Storage Space	4GB	8GB or more

Table 2: Hardware Requirement for Server

PC	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (4 Mbps)	Cable, Wi-Fi (8 Mbps)
Operating System	Window 7	Window 10
Computer Processor	Intel® Core i3 1.4GHz	Intel® Core i5 2.50GHz
Computer Memory	4GB RAM	8GB RAM or more
Storage Space	8GB	16GB or more
Web Browser	Chromes (v51 or higher)	Chrome latest stable version

Table 3: Hardware Requirement for PC.

1.3.5.2 Software requirements

Name	Name/Version	Description
Environment	Java JDK 8, Node v10	Specification for developing web server
Package manager	npm v6	Package manager for JavaScript Programming
Operating System	Window 10	Operating system and platform for development
Integrated development environment	- IntelliJ IDEA 2019.1.1 - Visual Studio Code	Programming tools
Modeling tool	draw.io, lucidchart	Software modeling tool
Database Management System	MySQL 5.7	Used to create & manage the database for system
Temporary Storage	Redis (2.8.6 or higher version)	Used to store data temporarily and load balancing data of MySQL
Source control	Github.com	Used to source control & version control.
Web browser	Chrome 76	Testing browser
Deployment server	Apache Tomcat 8	Deployment Environment

Table 4: Software requirement.

2. Project organization

2.1 Software Process Model

As the project problem which we mention above, we choose Scrum Model to manage project development with each sprint is 5 working days. This model will help us:
- Scrum team always work together to reach the project goal. Team member working progress is daily updated.
- Minimize risks when changing requirement.
- The working cycle "Learn - Revise – Apply" is done concurrently and continuously, a team member can apply new research and get the result quickly.
- Product coding is focused rather than document to deliver higher product value.
More information: https://www.scrum.org/

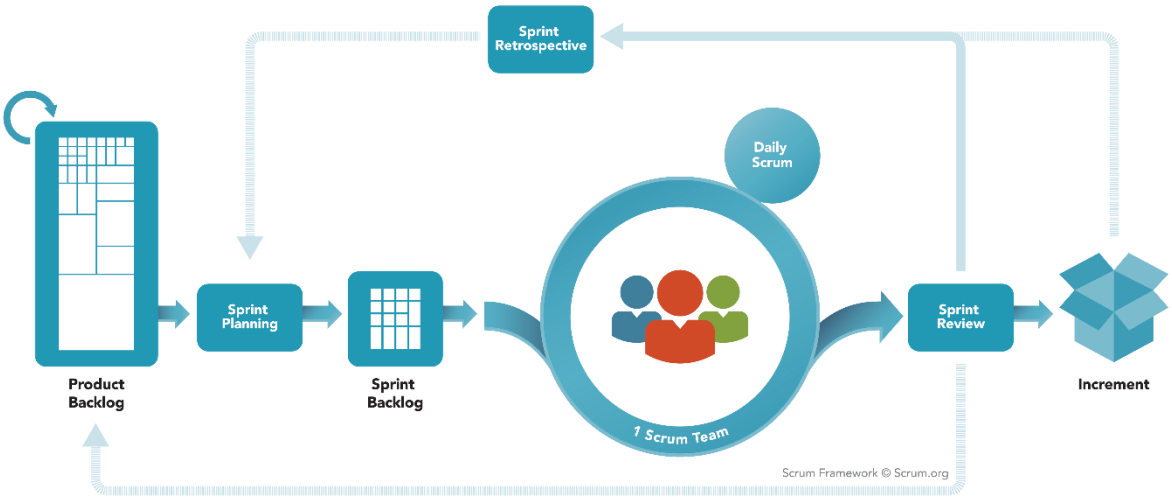


Figure 1: Scrum Model

2.2 Roles and responsibilities

No	Full name	Role in Group	Responsibilities
1	Nguyễn Huy Hùng	Project manager	• Specify user requirement • Control the development process • Give out technique and business analysis support

2	Phạm Hoàng Tuyết Ngân	Team Leader, BA, DEV, Tester	• Managing process • Designing database • Clarifying requirements • Prepare documents • GUI Design • Create test plan • Coding • Testing
3	Lưu Duy Hòa	Team member, BA, DEV, Tester	• Designing database • Clarifying requirements • Prepare documents • GUI Design • Create test plan • Coding • Testing
4	Đặng Văn Hoàng	Team member, BA, DEV, Tester	• Designing database • Clarifying requirements • Prepare documents • GUI Design • Create test plan • Coding • Testing

Table 5: Roles and Responsibilities Details

2.3 Tools and Techniques

Tools		
Developing tools	Front-end	Visual Studio Code
	Back-end	IntelliJ IDEA
Database		MySQL Workbench 8.0
Temporary Storage		Redis Cloud

Table 6: Tools

Techniques	
Front-end	HTML5, CSS3, Javascript ES6, jQuery 3.4, VueJS 2.0, Vuetify 2.0, Stockfishjs 10, Chessjs, Chessboardjs 1.0, Chessground 7.2
Back-end	JavaEE, Spring Framework 5.0, Spring Boot 2.0
Database	MySQL 5.7
Temporary Storage	Redis

Table 7: Techniques

3. Project Management Plan

3.1 Product Backlog

All product backlogs could be found here.

3.2 Sprint Backlog

All sprint backlogs could be found here.

3.3 Meeting Minutes

All meeting documents could be found here.

4. Coding Convention

We also use coding convention follow with programming language convention:

Java: Using for web server development

- Naming Convention:
  + **Camel-case** is used for the name of variables, methods, interfaces and classes. Interfaces and Classes name follow **Camel-case** with the uppercase first letter.
  + Name of methods should be a verb.
  + Constants name is a noun which is all uppercase and separated by an underscore.
- Indentation:
  + Using 8 spaces for the beginning of higher-level code.
  + Using 4 spaces for each break after a comma or a dot or break before an operator.
- Comment:
  + Single line of comment is used with //.
  + Block of comments is used with /**/.
- Statement:
  + Each line contain at most one statement and end with a semicolon.
  + if, for and while statement should contain bracket although perform a single task.
- White Space:
  + Blank line should be used to separate blocks of code which perform the same task.
  + Blank space should be used to separate variable and operator except for some special operator (increment, decrement,…).

Reference:
http://www.oracle.com/technetwork/java/codeconvtoc-136057.html

JavaScript ES6: Using for web front-end application development

The coding convention for JavaScript is the same with above convention and there are some different notes.

- Reference:
  + `const` is used for all of the references.
  + If reference is re-assign, let use `let`.
- String:
  + String should be used with a single quote.
- Statement:
  + Each line contain at most one statement and end without semicolon.

Reference: https://github.com/airbnb/javascript

C. Software Requirement Specification

1. User Requirement Specification

1.1 Guest Requirement

Guest is a user who does not access to the system. Guest can use some functions in the system, the guest should sign in to use more features. These are some functions guest can use:

- Sign in with Social Network (especially is Gmail)
- Sign up with Social Network (especially is Gmail)
- View published course
- View course details
- Search courses by name
- Filter courses by category and skill level

1.2 Authorized user Requirement

Authorized user is a user who accesses to the system as any role. These are some functions authorized user can use:

- Sign out
- View profile
- Edit profile

1.3 Learner Requirement

Learner is a person who user to the system as learner role. A learner can use the following functions:

- View published course
- View course details
- Search courses by name
- Filter courses by category and skill level
- View list of suggested course
- Enrol course
- View enrolled course
- Learn course by read interactive lesson
- Learn course by read uninteractive lesson
- Learn course by do exercise
- Manage course review (create or edit review)
- Remove course review
- View learning progress
- Play chess with the bot

1.4 Instructor Requirement

Instructor is a user who accesses to the system as instructor role. An instructor can use the following functions:

- View course reports
- Manage course (create, edit course or edit course status)
- Create or edit interactive lesson for building course content
- Create or edit uninteractive lesson for building course content
- Create or edit exercise for building course content
- Remove lesson or exercise for building course content

1.5 Admin Requirement

Admin is a person who accesses to the system as admin role. Admin can use the following functions:

- View course reports
- View chess bot reports
- View user reports
- View published course
- Unpublish course
- View user list
- Active or Deactivate learner and instructor
- Review new instructor
- Review new course
- Manage category (create or edit the category)
- Remove category

1.6 System Bot Requirement

System bot is an actor which auto perform functions. In this system, system bot can do:

- Play chess with learner
- Update learner skill level
- Make course suggestion for learner

2. System Requirement Specification

2.1 External Interface Requirement

2.1.1 User Interface

- The user interface uses Vietnamese.

- The user interface displays best on 1366x768 resolutions on desktop.

2.1.2 Hardware Interface

Desktop PC.

2.1.3 Software Interface

Web application: work with browsers Chromes (v51 or higher),or with any web browser that supports HTML5/CSS3 and ES6.

2.1.4 Software Interface

Use HTTP protocol 1.1 for communication between the web browser and the web.

2.2 System Overview Use case[1]

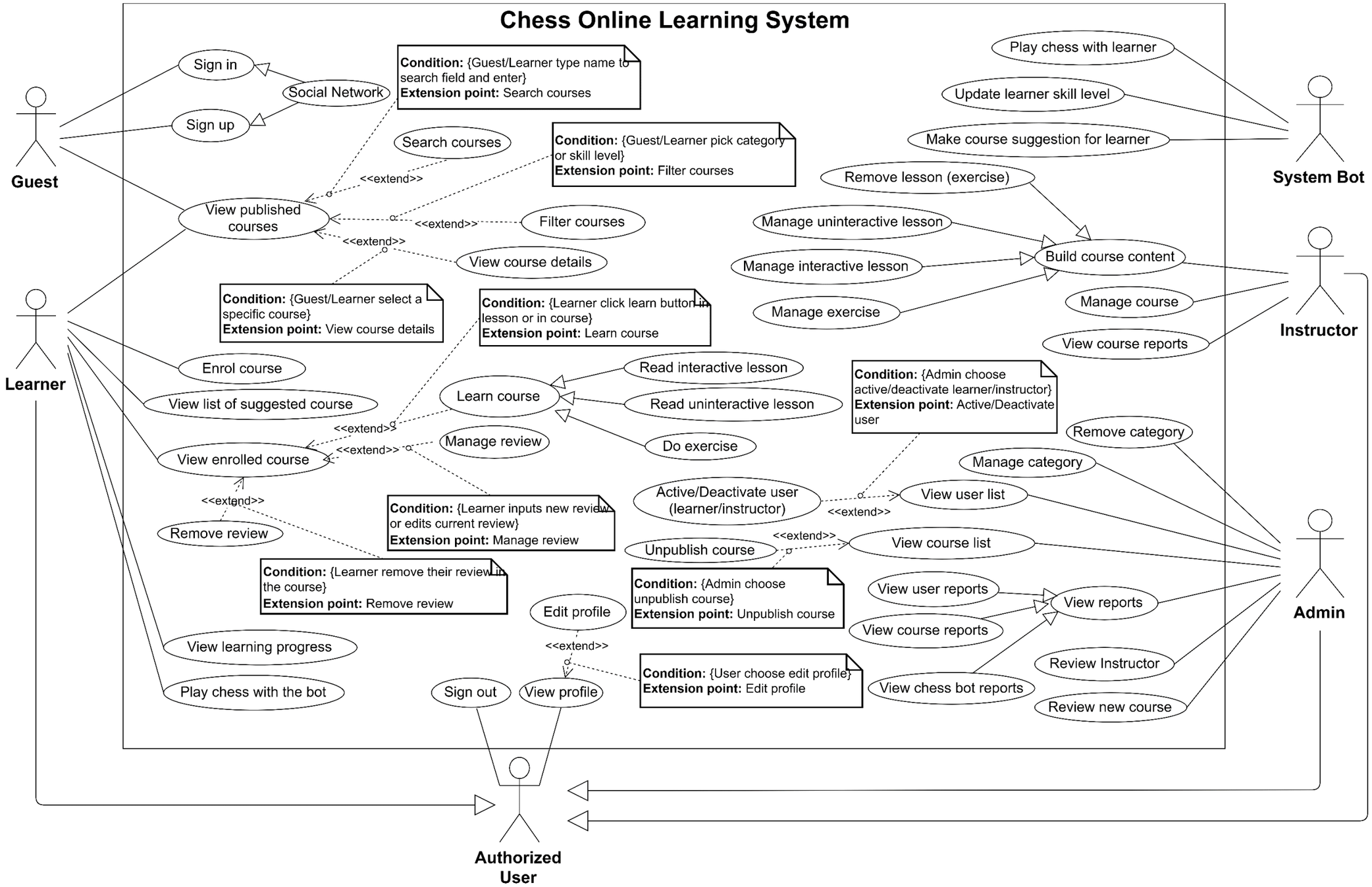


Figure 2: System Overview Use Cas

2.3 List of Use Case

2.3.1 <Guest> Overview Use Case

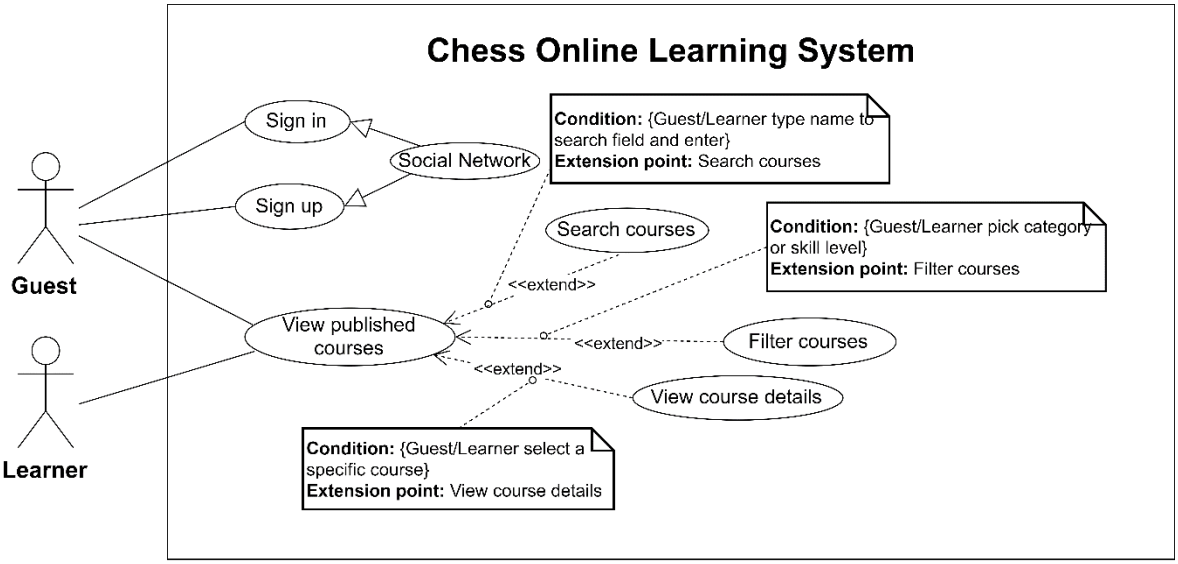


Figure 3: <Guest> Overview Use Case

2.3.1.1 <Guest> Sign In

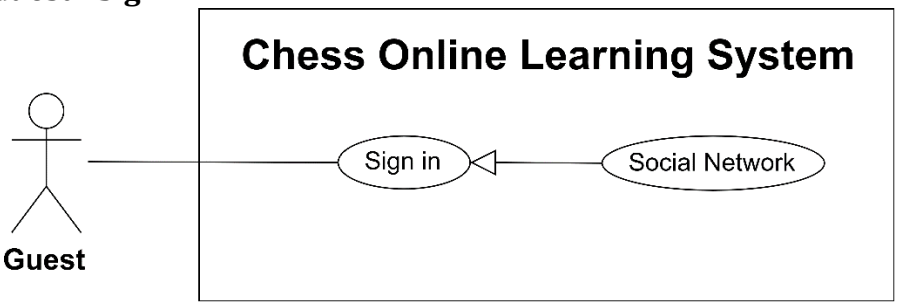


Figure 4: <Guest> Sign In

USE CASE – UC_GU.01			
Use Case No.	UC_GU.01	**Use Case Version**	2.0
Use Case Name	Sign In		
Author	Đặng Văn Hoàng		
Date	20/07/2019	**Priority**	Normal

Actor:

- Guest

Summary:

- This use case allows the guest to sign in the system.

Goal:

- Guest can sign in the system with a specific role.

Triggers:

- Guest sends a sign-in command.

Preconditions:

- Guest must have a Google account.

Postconditions:

- Success: Guest is authorized with a specific role in the system.

- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Guest clicks on "Đăng nhập" button.	System redirects guest to Google sign-in screen.
2	Guest inputs Google account or chooses a current signed-in Google account.	
3	Guest sends a sign-in command to the system.	Guest signs in the system with a specific role. [Exception 1]

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	Guest signs in to the wrong application.	System returns to current page without changing.

Relationships: N/A

Business Rules:

- User clicks on the signed-in button. The authorization request is sent to Google's OAuth servers.
- After sign in, an access token will return to the client.
- Base on the token, the system will check if the role is accepted or not based on each application:
  - If the role is "Learner" or "Registration", the guest must use the learning website.
  - If the role is "Instructor" or "Admin", the guest must use the admin website.

A new session is created in the server for the user.

Table 8: Sign In specification

2.3.1.2 <Guest> Sign Up

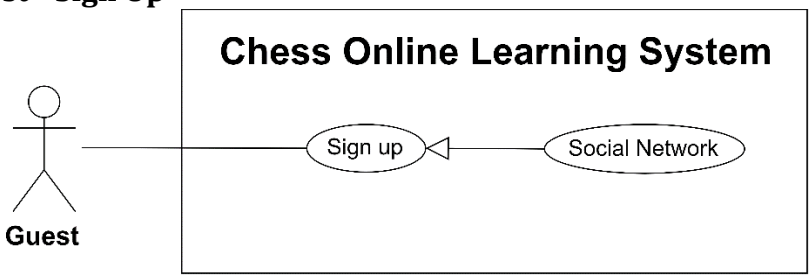


Figure 5: <Guest> Sign Up

USE CASE – UC_GU.02			
Use Case No.	UC_GU.02	**Use Case Version**	2.0
Use Case Name	Sign Up		
Author	Đặng Văn Hoàng		

Date	20/07/2019	Priority	Normal

Actor:

- Guest

Summary:

- This use case allows the guest to sign up as a learner or instructor.

Goal:

- Guest becomes a learner or instructor.

Triggers:

- Guest sign in the system for the first time.

Preconditions:

- Guest must have a Google account.

Postconditions:

- Success: New learner or instructor is added in the system.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Guest clicks on "Đăng nhập" button.	The system confirms new user and display register form with information: • Ảnh đại diện: file field • Email: text field • Họ và tên: text field • Học viên (auto-selected), Người hướng dẫn: radio button • Trình độ: select box (for learner only) • Xác nhận điều khoản: checkbox • Xác nhận: button System required information: • Họ và tên: min length 6, max length 255.
2	Guest edit their name and avatar.	[Exception 1] [Exception 2]
3	Guest input required learner information. [Alternative 1]	
4	Guest click "Xác nhận điều khoản".	The system will record user has been accepted all rules in our system. [Exception 3]
5	Guest click "Xác nhận" button.	The system redirects to Home page and shows a message: • For learners: "Chúc mừng bạn đã đăng ký trở thành học viên thành công.".

		• For learners: "Chúc mừng bạn đã đăng ký trở thành người hướng dẫn thành công. Vui lòng chờ hệ thống xét duyệt tài khoản của bạn.".

Alternative Scenario:

No	Actor Action	System Response
1a	Guest clicks "Người hướng dẫn" button.	System show form register instructor: • Thành tích: text area • Các chứng nhận: multiple upload image
1b	Guest input required instructor information.	

Exceptions:

No	Cause	System Response
1	Guest upload file not image format.	System shows error messages "Hình ảnh không đúng định dạng".
2	New guest name is incorrect with name rules.	System shows errors messages for each rule: • Name is empty: "Họ và tên không được bỏ trống" • Name is less than 6 characters: "Họ và tên không được ít hơn 6 kí tự" • Name is more than 225 characters: "Họ và tên không được nhiều hơn 225 kí tự"
3	Guest do not click "Xác nhận điều khoản".	System shows error messages: "Bạn phải đồng ý để tiếp tục".

Relationships: N/A

Business Rules:

- User click "Lưu" button, the system will upload all images on 3rd party and get image URL.
- System will add new account to the database.
- If a new learner signed up successfully, the skill level will convert to an initial point for the learner.
- If user click "Đăng nhập" button but he does not perform sign-up case, the database will store user with "Registration" role.
- If a user has been stored in the database with "Registration" role, system auto redirect to Sign-up page after he clicks "Đăng nhập".

Table 9: Sign Up specification

2.3.1.3 <Guest & Learner> View Published Courses

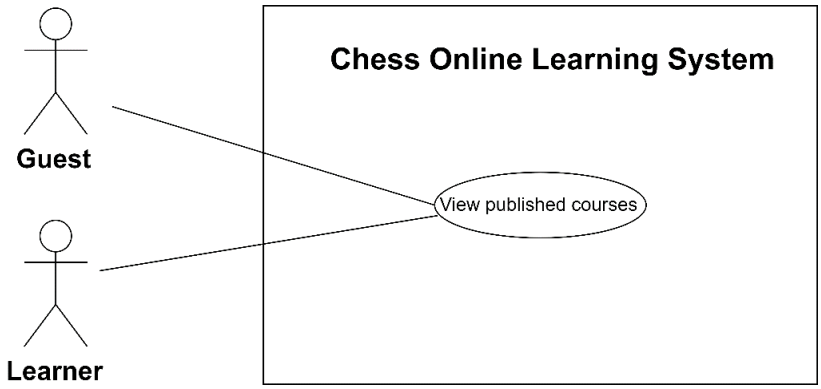


Figure 6: <Guest & Learner> View Published Courses

USE CASE – UC_GU.03			
Use Case No.	UC_GU.03	**Use Case Version**	2.0
Use Case Name	View Published Courses		
Author	Lưu Duy Hòa		
Date	20/07/2019	**Priority**	High

Actor:

- Guest & Learner

Summary:

- This use case allows the actor view list of published course.

Goal:

- Guest or Learner can view the list of published course in the system.

Triggers:

- View published course list command is sent.

Preconditions:

- N/A

Postconditions:

- Success: System shows a list of published course.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	On the home page, actor clicks on "Bắt đầu học" to view all published course in the system.	The system will direct to the course overview page and show all published courses in the system with basic information for each course: name, description, required skill level, instructor name, categories, rating and course image.

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: • The system will show all course which has published for guests and learners.

Table 10: View Published Courses specification

2.3.1.4 <Guest & Learner> View Course Details

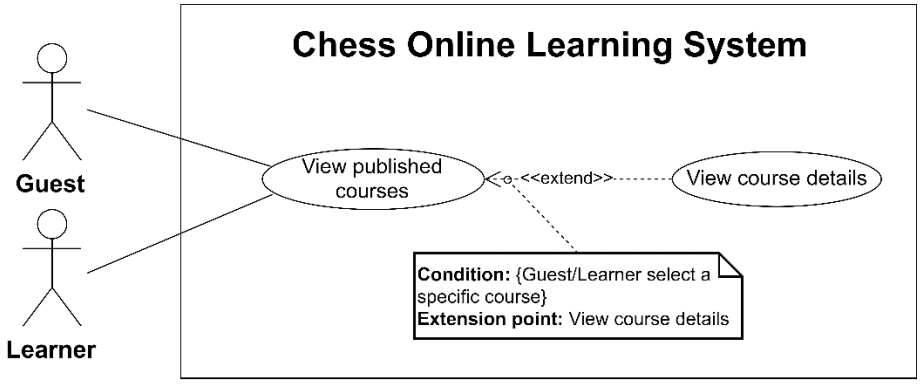


Figure 7: <Guest & Learner> View Course Details

USE CASE – UC_GU.04			
Use Case No.	UC_GU.04	**Use Case Version**	2.0
Use Case Name	View Course Details		
Author	Lưu Duy Hòa		
Date	20/07/2019	**Priority**	High

Actor:
- Guest & Learner

Summary:
- This use case allows the guest or learner to view course details.

Goal:
- Guest or learner can view details of a specific course.

Triggers:
- Actor selects a course from the published list.

Preconditions:
- N/A

Postconditions:
- Success: System shows details of the selected course.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor clicks on course to view details of them.	The system shows detail course include: • "Giảng viên": instructor information • "Số lượng học viên": number of enrolled learners • "Số lượng bài học": number of the lesson

		• "Giáo trình": list of the lesson • Thông tin: course description • "Cấp độ": required skill level • "Điểm tối thiểu": required point • "Đánh giá": Overview box and the latest comment box • "Gợi ý khóa học": list of suggested courses (for the learner) • "Đăng ký khóa học": enrolled button (for the learner)

Alternative Scenario: N/A

Exceptions: N/A

Relationships: UC_GU.03

Business Rules:

- The system will return details of the selected course.
- Learner can enrol and learn course from here.
- Course reviews are filtered by rating number.

Table 11: View course details specification

2.3.1.5 <Guest & Learner> Search Courses

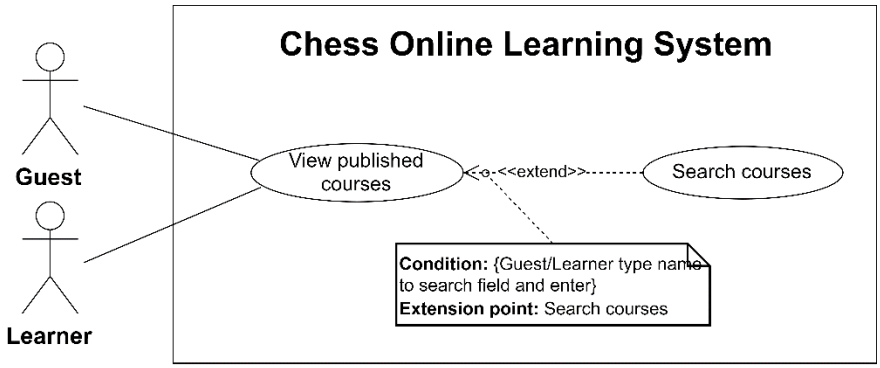


Figure 8: <Guest & Learner> Search Courses

USE CASE – UC_GU.05			
Use Case No.	UC_GU.05	**Use Case Version**	2.0
Use Case Name	Search Courses		
Author	Đặng Văn Hoàng		
Date	20/07/2019	**Priority**	High

Actor:

- Guest & Learner

Summary:

- This use case allows the actor can search courses.

Goal:

- Guest or learner can search courses by course's name.

Triggers:

- Actor type name of the course and send search command.

Preconditions:

- N/A

Postconditions:

- Success: System shows list of course which contains search keyword.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor inputs search keyword in text input.	[Exception 1]
2	Actor clicks on "Tìm Kiếm" button.	System finds courses by keyword at search text input in the database and returns the result.

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	Actor clicks on "Tìm kiếm" button with empty text input.	System shows an error message "Tên khóa học không được để trống".

Relationships: UC_GU.03

Business Rules:

- When actor sends search command, system performs search published course by name in the database and return the result.
- System only supports search courses by course name.

Table 12: Search Courses specification

2.3.1.6 <Guest & Learner> Filter Courses

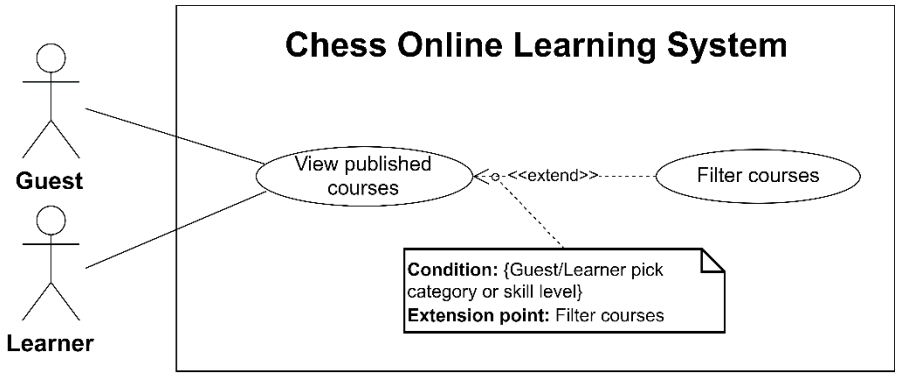


Figure 9: <Guest & Learner> Filter Courses

USE CASE – UC_GU.06			
Use Case No.	UC_GU.06	**Use Case Version**	2.0
Use Case Name	Filter Courses		
Author	Đặng Văn Hoàng		
Date	20/07/2019	**Priority**	High
Actor:			

- Guest & Learner

Summary:

- This use case allows the actor to filter courses by category or skill level.

Goal:

- Guest or learner can filter courses by conditions.

Triggers:

- Actor picks condition to filter.

Preconditions:

- N/A

Postconditions:

- Success: System shows a list of course which was filtered by the condition.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor clicks on category or skill level which need to be filtered.	System finds and shows a list of course which was filtered by conditional.

Alternative Scenario: N/A

Exceptions: N/A

Relationships: UC_GU.03

Business Rules:

- Guest or Learner can view the list of course which was filtered by category or skill level.

Table 13: Filter Courses specification

2.3.3 <Learner> Overview Use Case

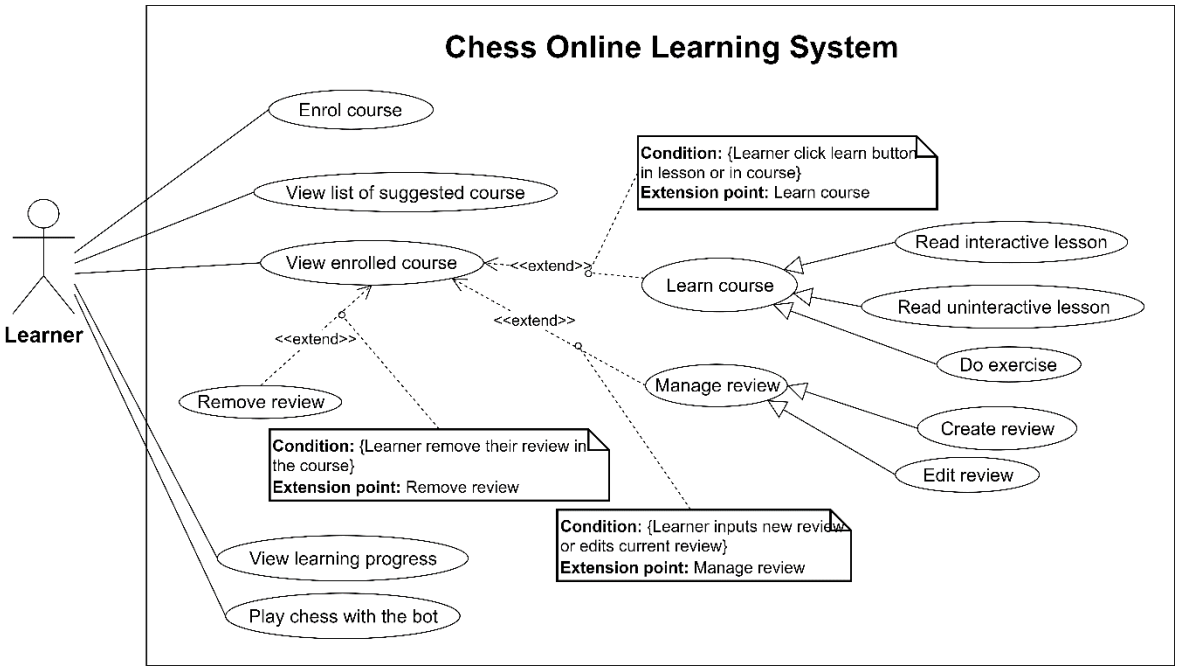


Figure 10: <Learner> Overview Use Case

2.3.3.1 <Learner> Enrol Course

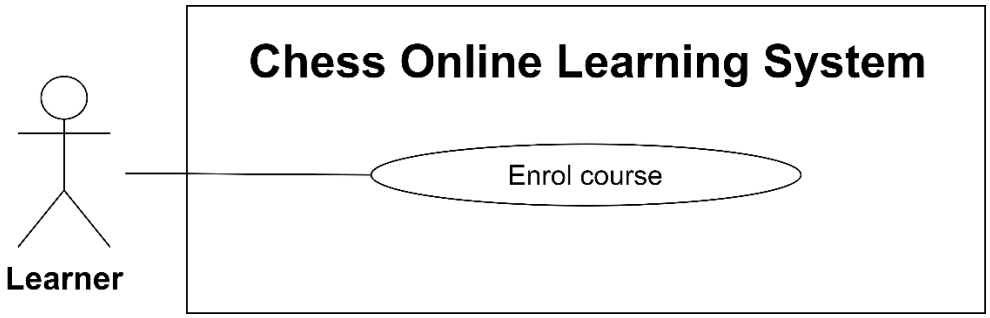


Figure 11: <Learner> Enrol Course

USE CASE – UC_LE.01			
Use Case No.	UC_LE.01	**Use Case Version**	2.0
Use Case Name	Enrol Course		
Author	Lưu Duy Hòa		
Date	20/07/2019	**Priority**	High

Actor:

- Learner

Summary:

- This use case allows the learner to enrol system course.

Goal:

- Learner can enrol to learn a specific course under their skill level.

Triggers:

- Learner sends the enrol course command.

Preconditions:

- User must sign in into the system with role Learner.

Postconditions:

- Success: System displays enrol course success messages.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Learner navigates to "Đăng ký khóa học" from the course details page.	System shows confirmation popup: • "Xác nhận": button • "Hủy": button
2	Learner click "Xác nhận" button on the confirmation popup. [Alternative 1]	System closes the confirmation popup and shows a message "Đăng kí khóa học thành công". System unlocks lesson list. The "Đăng ký khóa học" button is changed to "Bắt đầu học" button. [Exceptions 1]

Alternative Scenario:

No	Actor Action	System Response

1	Learner clicks on "Hủy" button.	System closes confirmation popup.

Exceptions:

No	Cause	System Response
1	The Learner has not enough required skill level to enrol.	System shows an error message "Bạn chưa đủ điều kiện để đăng kí khóa học này. Xin thử lại các khóa học khác!".

Relationships: N/A

Business Rules:

- System will show the enrol button for learners in the course details page.
- Courses in the system require a skill level to enrol.
- A learner can only enrol courses whose skill level is lower or equal with Learner skill level.

Table 14: Enrol Course specification

2.3.3.2 <Learner> View Enrolled Course

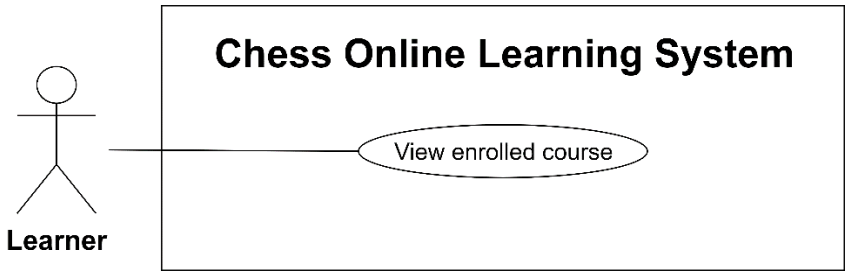


Figure 12: <Learner> View Enrolled Course

USE CASE – UC_LE.02			
Use Case No.	UC_LE.02	**Use Case Version**	2.0
Use Case Name	View Enrolled Course		
Author	Lưu Duy Hòa		
Date	20/07/2019	**Priority**	High

Actor:

- Learner

Summary:

- This use case allows the learner can view an enrolled course.

Goal:

- Learner can view a specific course which they enrolled.

Triggers:

- Learner sends view enrolled course command.

Preconditions:

- User must sign in into the system with role Learner.
- Learner had enrolled in the course.

Postconditions:

- Success: System displays enrolled course page.

- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Learner navigates to a course which they enrolled.	System shows course details page and enables review area, unlock lessons.

Alternative Scenario: *N/A*

Exceptions: N/A

Relationships: N/A

Business Rules:

- Learner can learn course or learn a specific lesson in enrolled courses.
- Learner can review course which is enrolled.

Table 15: View Enrolled Course specification

2.3.3.3 <Learner> Create Review

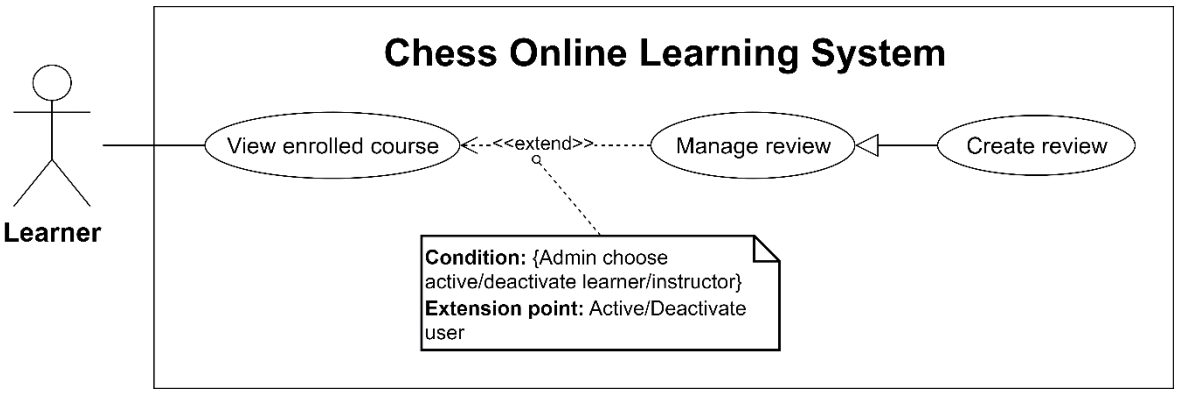


Figure 13: <Learner> Create Review

USE CASE – UC_LE.03			
Use Case No.	UC_LE.03	**Use Case Version**	2.0
Use Case Name	Create Review		
Author	Lưu Duy Hòa		
Date	20/07/2019	**Priority**	Normal

Actor:

- Learner

Summary:

- This use case allows the learner to create a review to course.

Goal:

- User can review a specific course.

Triggers:

- Learner sends the command to review the course.

Preconditions:

- User must sign in into the system with role Learner.
- Learner had enrolled in the course.

Postconditions:

- Success: System add new review for course successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	On the course details page, learner moves to review area.	System highlights the review area: • Đánh giá của bạn: rating • Bình luận: text field • Đăng: button System required information: • Rating must be chosen
2	Learner clicks "rating" and input text field.	[Exceptions 1]
3	Learner clicks "Đăng" button. [Alternative 1]	System re-calculates course review and displays new review on the top of list. [Exception 2]

Alternative Scenario:

Step	Actor Action	System Response
1	Learner removes his action.	There is no review added.

Exceptions:

Step	Cause	System Response
1	Learner does not rate review.	System shows the error message "Điểm đánh giá phải lớn hơn 0".
2	Learner has reviewed on this course before.	System show error message "Đánh giá không thành công. Bạn đã đánh giá khóa học này rồi".

Relationships: UC_LE.02

Business Rules:

- Learner can only review when enrolled course.
- Each learner can only have 1 review in each course.
- Learner review will be shown on the top of Review list.
- Course review is re-calculated.

Table 16: : Create Review specification

2.3.3.4 <Learner> Edit Review

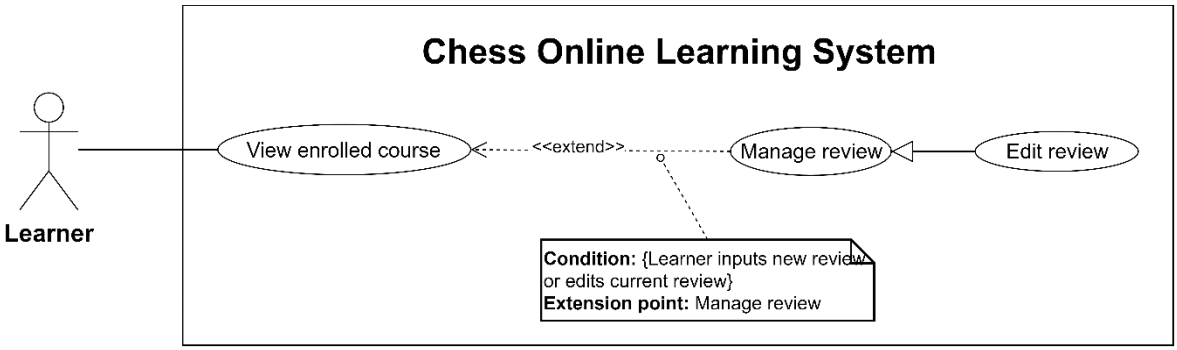


Figure 14: <Learner> Edit Review

USE CASE – UC_LE.04			
Use Case No.	UC_LE.04	**Use Case Version**	2.0
Use Case Name	Edit Review		
Author	Lưu Duy Hòa		
Date	20/07/2019	**Priority**	Normal

Actor:

- Learner

Summary:

- This use case allows the learner can edit their review.

Goal:

- Learner can edit his review in the course.

Triggers:

- Learner sends the command to edit the course review.

Preconditions:

- User must sign in into the system with role Learner.
- Learner had enrolled in the course.
- Learner used to review in the course.

Postconditions:

- Success: System updates course review successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	On their course review, learner clicks on "Sửa" button.	System highlights the rating frame and comments: • Đánh giá của bạn: rating • Bình luận: text field • Đăng: button System required information: • Rating must be chosen
2	Learner clicks "rating" and input to the text field.	[Exceptions 1]

3	Learner clicks "Lưu" button [Alternative 1]	System display updated review successfully.

Alternative Scenario:

Step	Actor Action	System Response
1	Learner removes his action.	There is no review added.

Exceptions:

Step	Cause	System Response
1	Learner rates to 0.	System shows the error message "Điểm đánh giá phải lớn hơn 0".

Relationships: UC_LE.02

Business Rules:

- Learner can edit his course review.
- Learner review is displayed on the top of Review list.
- Course review is re-calculated.

Table 17: Edit Review specification

2.3.3.5 <Learner> Remove Review

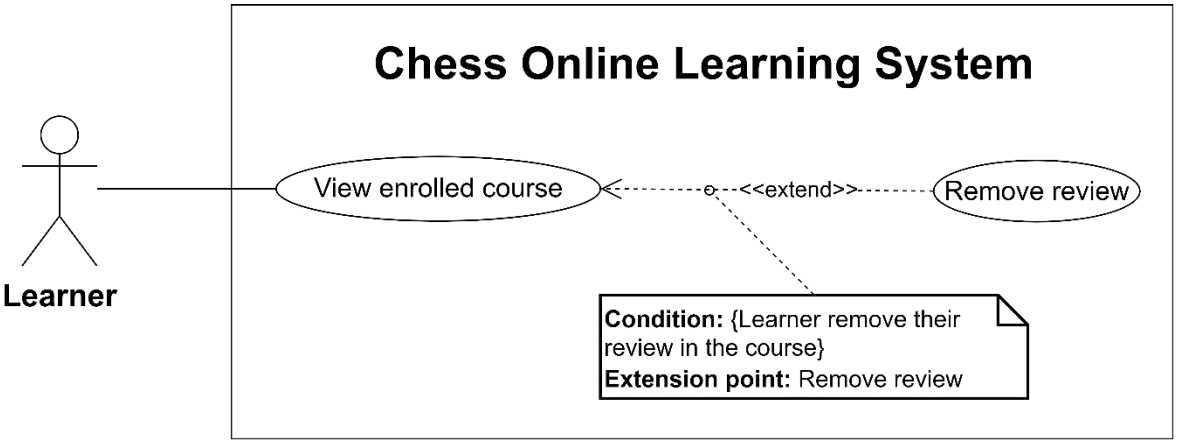


Figure 15: <Learner> Remove Review

USE CASE – UC_LE.05			
Use Case No.	UC_LE.05	**Use Case Version**	2.0
Use Case Name	Remove Review		
Author	Lưu Duy Hòa		
Date	20/07/2019	**Priority**	Normal

Actor:

- Learner

Summary:

- This use case allows the learner can remove their review.

Goal:

- Learner can remove his review in the course.

Triggers:

- Learner sends the command to remove the course review.

Preconditions:

- User must sign in into the system with role Learner.
- Learner had enrolled in the course.
- Learner must review in the course.

Postconditions:

- Success: System removes course review successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	On their course review, learner clicks "Xóa" button.	System displays confirm dialog: • Bạn có muốn xóa bình luận này: text • Xác nhận: button • Đóng: button
2	Learner clicks on "Xác nhận" button. [Alternative 1]	The review is removed and the system displays a message "Xóa bình luận thành công".

Alternative Scenario:

Step	Actor Action	System Response
1	Learner clicks on "Đóng" button.	System closes the dialog and there is no changing.

Exceptions: N/A

Relationships: UC_LE.02

Business Rules:

- Learner can remove his review in the course.
- System calculates course review again.

Table 18: Remove Review specification

2.3.3.6 <Learner> Read Interactive Lesson

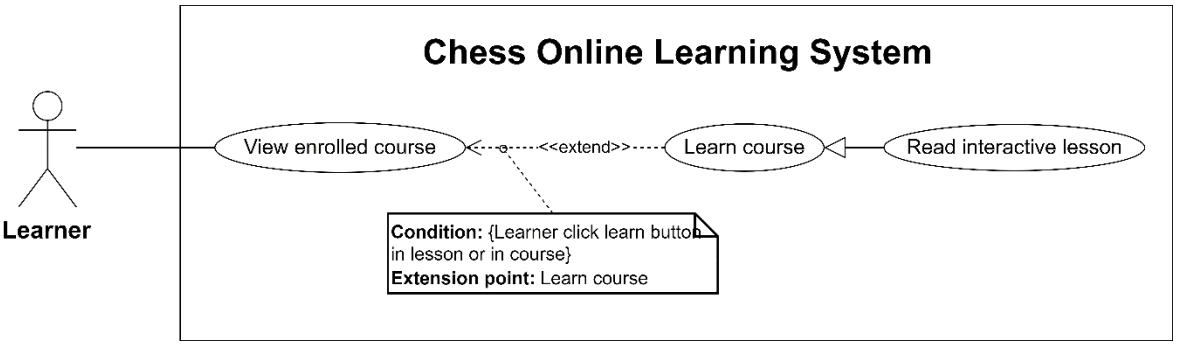


Figure 16: <Learner> Read Interactive Lesson

USE CASE – UC_LE.06			
Use Case No.	UC_LE.06	**Use Case Version**	2.0
Use Case Name	Read Interactive Lesson		
Author	Lưu Duy Hòa		
Date	20/07/2019	**Priority**	High

Actor:

- Learner

Summary:

- This use case allows the learner to read the interactive lesson.

Goal:

- Learner can read an interactive lesson.

Triggers:

- Learner sends the command to read the interactive lesson.

Preconditions:

- User signed in to the system as a learner successfully.
- Learner had enrolled in the course.

Postconditions:

- Success: Learner can read lesson content.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Learner clicks on "Bắt đầu học" button. [Alternative 1]	System redirects learner to the next lesson which learner did not learn. If the type is interactive lesson, system shows the lesson content: • Chessboard • Move arrow button group • Move content
2	Learner can click on the move or arrow buttons. [Alternative 2]	System shows the current chess position following the move and may be have a move content.
3	Learner clicks "Bài tiếp" button to continue for the next lesson. [Alternative 3]	System record that learner has finished this lesson and move to the next lesson.

Alternative Scenario:

Step	Actor Action	System Response
1	Learner clicks on a specific lesson.	System redirects to the selected lesson.
2	Learner clicks on the exit button to back to course detail.	System redirects learner to course detail.

3	Learner clicks on "Bài trước" to go to previous lesson.	System shows previous detail lesson. The lesson is not logged to finish.

Exceptions: N/A
Relationships: UC_LE.02
Business Rules:

- Interactive lesson is a lesson which contains a chess-game is analysed. All chess-game moves and position following the move will be loaded and learner learns the lesson by view chess position with a specific move.
- Interactive lesson is recorded to finish when learner request next lesson.
- Course is finished when learner finish all lessons and exercises in the course.

Table 19: Read Interactive Lesson specification

2.3.3.7 <Learner> Read Uninteractive Lesson

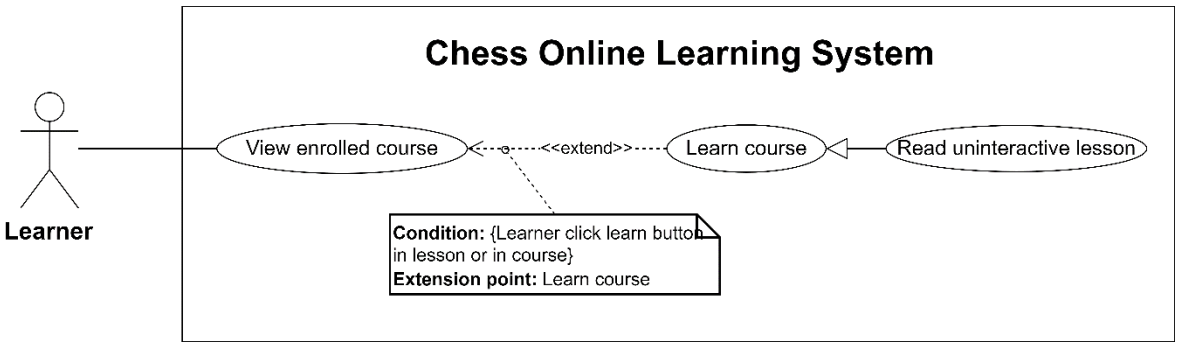


Figure 17: <Learner> Read Uninteractive Lesson

USE CASE – UC_LE.07			
Use Case No.	UC_LE.07	**Use Case Version**	2.0
Use Case Name	Read Uninteractive Lesson		
Author	Đặng Văn Hoàng		
Date	23/03/2013	**Priority**	High

Actor:

- Learner

Summary:

- This use case allows the learner to read the uninteractive lesson.

Goal:

- Learner can read an uninteractive lesson.

Triggers:

- Learner sends the command to read the uninteractive lesson.

Preconditions:

- User must sign in into the system with role Learner.
- Learner had enrolled in the course.

Postconditions:

- Success: Learner can read the uninteractive lesson content.

- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Learner clicks on "Bắt đầu học" button. [Alternative 1]	System redirects learner to the next lesson which learner did not learn.
2	If lesson type is uninteractive, the learner can read lesson content. [Alternative 2]	
3	Learner clicks "Bài tiếp" button to continue for the next lesson. [Alternative 3]	System record that learner has finished this lesson and move to the next lesson.

Alternative Scenario:

Step	Actor Action	System Response
1	Learner clicks on a specific lesson.	System redirects to the selected lesson.
2	Learner clicks on the exit button to back to course detail.	System redirects learner to course detail.
3	Learner clicks "bài trước" to previous lesson.	System shows previous detail lesson. Lesson is not logged to finish.

Exceptions: N/A

Relationships: UC_LE.02

Business Rules:

- Uninteractive lesson content is text and media, which is a post from WYSIWYG Editor.
- Uninteractive is recorded to finish when learner request next lesson.
- Course is finished when the learner finishes all lessons and exercises in the course.

Table 20: Read Uninteractive Lesson specification

2.3.3.8 <Learner> Do Exercise

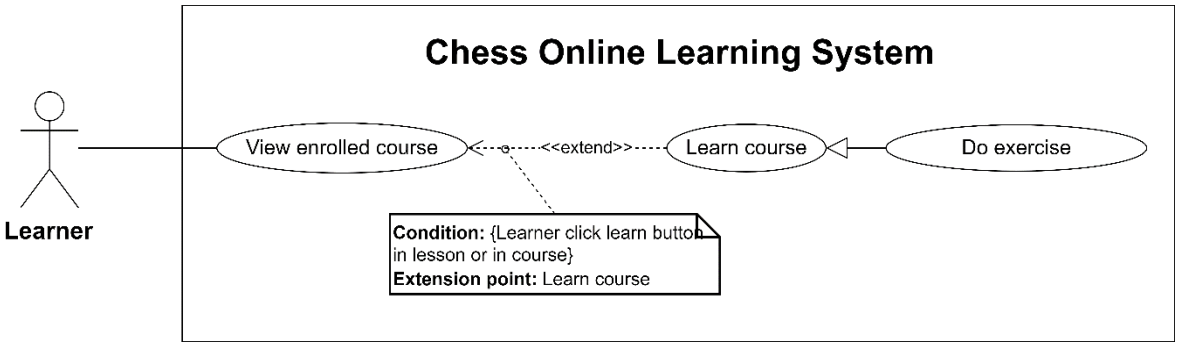


Figure 18: <Learner> Do Exercise

USE CASE – UC_LE.08			
Use Case No.	UC_LE.08	**Use Case Version**	2.0
Use Case Name	Do Exercise		
Author	Phạm Hoàng Tuyết Ngân		
Date	20/07/2019	**Priority**	High

Actor:

- Learner

Summary:

- This use case allows the learner to do exercise.

Goal:

- Learner can do exercise.

Triggers:

- Learner sends the do exercise command.

Preconditions:

- User signed in as a learner successfully.
- Learner had enrolled in the course.

Postconditions:

- Success: Learner can do exercise by performing their chess move.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Learner clicks on "Bắt đầu học" button. [Alternative 1]	System redirects learner to the next lesson which learner had not learned.
2	If lesson type is exercise, learner read the question and perform moves to solve the problem. [Alternative 2]	System checks learner move and sends a response for the wrong move or right move (it depends on the exercise content). If learner performs a right move, the system performs a move and change turn to learner again. This action is

		looped until learner solved the problem and finish the exercise. [Exception 1]
3	Learner clicks on "Bài kế tiếp" button to continue for the next lesson. [Alternative 3]	System record that learner has finished this exercise and move to the next lesson.

Alternative Scenario:

Step	Actor Action	System Response
1	Learner clicks on a specific lesson.	System redirects to the selected lesson.
2	Learner clicks on the exit button to back to course detail.	System redirects learner to course detail.
3	Learner clicks on "Bài trước" button to the previous lesson.	System shows the previous detail lesson. Lesson is not logged to finish.

Exceptions:

Step	Cause	System Response
1	Learner answers wrong.	System resets exercise and does not record this exercise is finished.

Relationships: UC_LE.02

Business Rules:

- Exercise is a lesson which provides a problem for the learner, learner must solve the problem to pass this lesson.
- In our system exercise will contain chess position and require learner must perform their chess move.
- Exercise is finished when learner solves the problem successfully.
- Exercise contains 2 types: Bot answer and Manual answer. In the Bot answer, a learner must solve the problem to checkmate. In the Manual answer, there are 1 to 3 answer and learner only need to solve right 1 answer.
- Course is finished when the learner finishes all lessons and exercises in the course.

Table 21: Do Exercise specification

2.3.3.9 <Learner> View Learning Progress

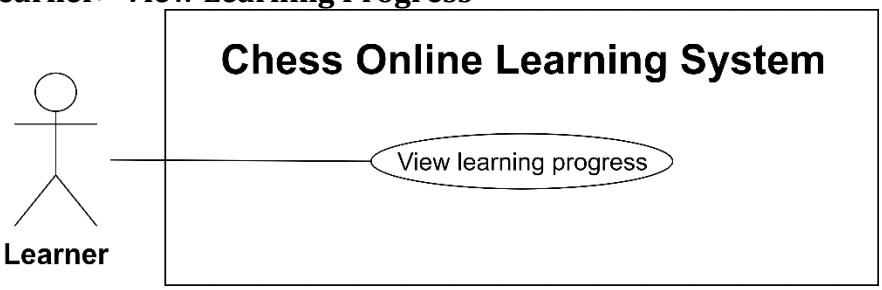


Figure 19: <Learner> View Learning Progress

USE CASE – UC_LE.09			
Use Case No.	UC_LE.09	**Use Case Version**	2.0
Use Case Name	View Learning Progress		
Author	Lưu Duy Hòa		
Date	20/07/2019	**Priority**	Normal

Actor:

- Learner

Summary:

- This use case allows the learner to view their learning progress.

Goal:

- Learner can view his learning progress.

Triggers:

- Learner sends the command to view learning progress.

Preconditions:

- User must sign in into the system with role Learner.
- Learner had enrolled in the course.

Postconditions:

- Success: System shows learner's courses and their progress.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Learner navigates to list of their enrolled course.	System responses their list of enrolled course with the percentage of finish in course.

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Learner can view their course list and progress for each course

Table 22: View Learning Progress specification

2.3.3.10 <Learner> View List Of Suggested Course

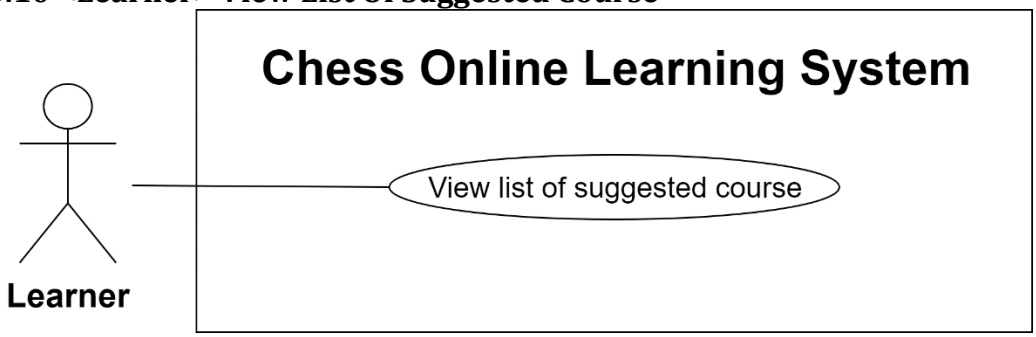


Figure 20: <Learner> View List Of Suggested Course

USE CASE – UC_LE.10			
Use Case No.	UC_LE.10	**Use Case Version**	2.0
Use Case Name	View List Of Suggested Course		
Author	Đặng Văn Hoàng		
Date	09/09/2019	**Priority**	High

Actor:

- Learner

Summary:

- This use case allows the learner to view the course suggested list.

Goal:

- Learner can his suggested courses.

Triggers:

- Learner sends the view course list or view course details command.

Preconditions:

- User signed in to the system as a learner successfully.

Postconditions:

- Success: System displays a suggested course list for the learner.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Learner navigates to suggested course list from the course overview page. [Alternative 1]	System calculates appropriate course for the learner from learner history and similar learners and displays to learner. [Exception 1]

Alternative Scenario:

Step	Actor Action	System Response
1	Learner views specific course details.	System displays appropriate courses for the learner based on the current course.

Exceptions:

Step	Cause	System Response

1	Learner has not learned any course.	Suggested list is empty.

Relationships: N/A
Business Rules:
- Learner can view his suggested course list to get an appropriate course.

Table 23: View List Of Suggested Course specification

2.3.3.11 <Learner> Play Chess With The Bot

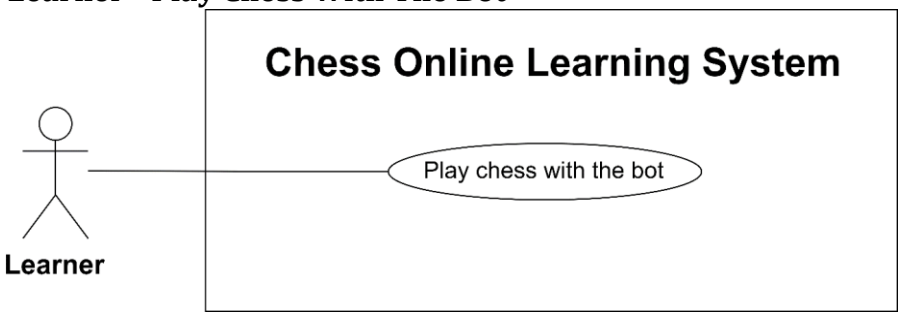


Figure 21: <Learner> Play Chess With The Bot

USE CASE – UC_LE.11			
Use Case No.	UC_LE.11	**Use Case Version**	2.0
Use Case Name	Play Chess With The Bot		
Author	Phạm Hoàng Tuyết Ngân		
Date	20/07/2019	**Priority**	High

Actor:
- Learner

Summary:
- This use case allows the learner to play chess with the bot.

Goal:
- Learner can play chess with the bot.

Triggers:
- Learner sends the playing with bot command.

Preconditions:
- User signed in to the system as a learner successfully.

Postconditions:
- Success: System starts the game and system bot will autoplay with the learner.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Learner navigates to playing game page.	System moves to the playing chess page: • Chessboard • Nước đi: move content-box

		• Chess game description: content-box • Ván đấu mới: button
2	Learner clicks on the new game button.	System shows setting game dialogue: • "Cấp độ": switch button • "Chọn màu quân": button group • "Thời gian đấu": time picker • "Bắt đầu": button • "Đóng": button System required information: • Game Time must larger than 1 minutes. • Game colour will be random if the learner does not choose.
3	Learner settings and clicks "Bắt đầu button. [Alternative 1]	System calculates win, draw and lose point for learner follow the Elo rating system. The game is started. [Exception 1]
4	Learner performs a move in the learner turn. [Alternative 2]	System gets the best move and performs move if the turn is bot. This action loop until the game is drawn or 1 player win. System record each move on the move content-box. When ending the game, system calculates a new point for the learner.

Alternative Scenario:

Step	Actor Action	System Response
1	Learner clicks "Đóng" button on setting game dialogue.	System closes the setting game dialogue. The game does not start.
2	Learner stops playing the game.	System record that learner has given up and lost the game.

Exceptions:

Step	Cause	System Response
1	Game Time less than 1 minutes.	The game cannot be started.

Relationships: N/A

Business Rules:

- Learner can play chess with system bot to get bonus point.
- Learner skill level can be changed after updating points.
- System will record all moves in the game.

Table 24: Play Chess With The Bot specification

2.3.4 <System Bot> Overview Use Case

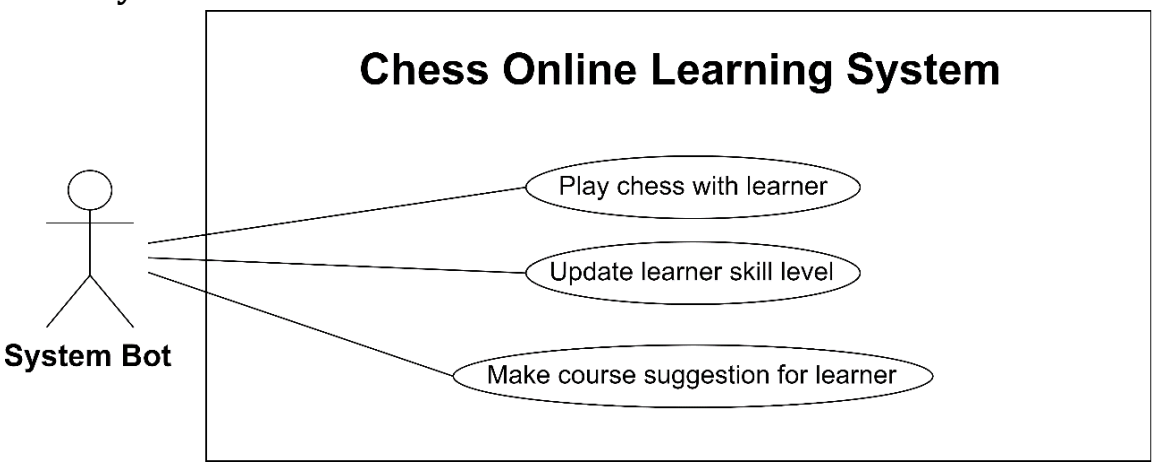


Figure 22: <System Bot> Overview Use Case

2.3.4.1 <System Bot> Play Chess With Learner

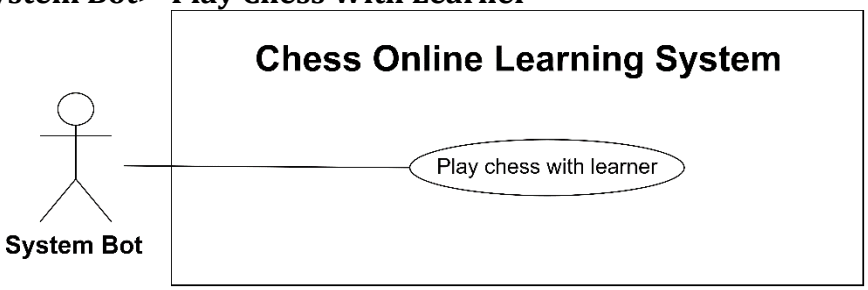


Figure 23: <System Bot> Play Chess With Learner

USE CASE – UC_SB.01			
Use Case No.	UC_SB.01	**Use Case Version**	2.0
Use Case Name	Play Chess With Learner		
Author	Phạm Hoàng Tuyết Ngân		
Date	20/07/2019	**Priority**	High

Actor:
- System Bot

Summary:
- This use case allows the system to autoplay chess with the learner.

Goal:
- System can autoplay chess.

Triggers:
- System sends the perform move commands when the turn is bot turn.

Preconditions: N/A

Postconditions:
- Success: New move is performed automatically.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	In bot turn, the system will analyse game position and calculate the best move. The new best move will be sent and performed in chessboard.	Move is performed and added to move content-box, turn is changed.

Alternative Scenario: N/A
Exceptions: N/A
Relationships: N/A
Business Rules:
- System can play chess with the learner.
- System bot is triggered if a user sends position, the system will analyse and get the best move.
- This use-case is used for playing chess with learner and response for the learner in exercise.

Table 25: Play Chess With Learner specification

2.3.4.2 <System Bot> Update Learner Skill Level

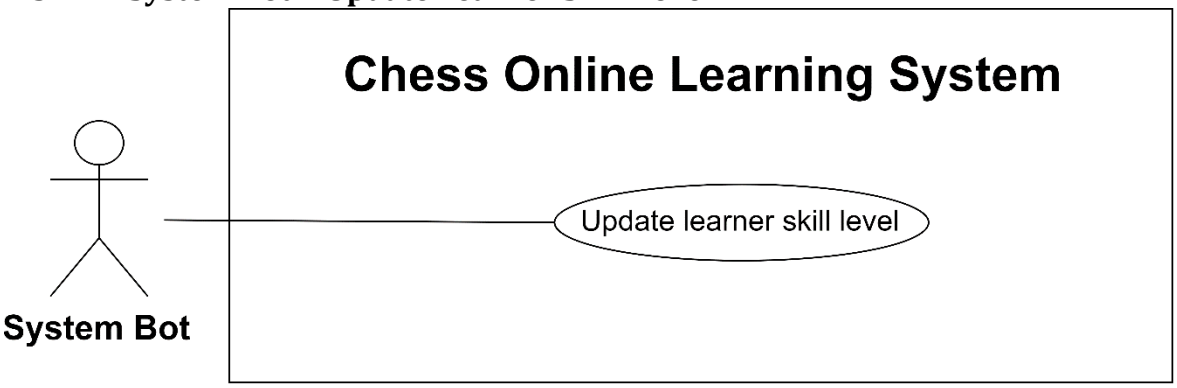


Figure 24: <System Bot> Update Learner Skill Level

USE CASE – UC_SB.02			
Use Case No.	UC_SB.02	**Use Case Version**	2.0
Use Case Name	Update Learner Skill Level		
Author	Phạm Hoàng Tuyết Ngân		
Date	09/09/2019	**Priority**	Normal

Actor:
- System Bot

Summary:
- This use case allows the system to auto-update chess skill level of the learner.

Goal:
- Learner skill level is auto-updated base on their point.

Triggers:
- Learner point is changed to another range of skill level.

Preconditions: N/A
Postconditions:
- Success: Learner skill level is updated.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Learner point is increased or decrease after the game is over and point is changed to another range of skill level.	Learner skill level is updated.

Alternative Scenario: N/A

Exceptions: N/A
Relationships: N/A
Business Rules:
- Learner skill level is specified by points.
- Each skill level has a range of point.
- System check learner skill level based on their points.

Table 26: Update Learner Skill Level specification

2.3.4.3 <System Bot> Make Course Suggestion For Learner

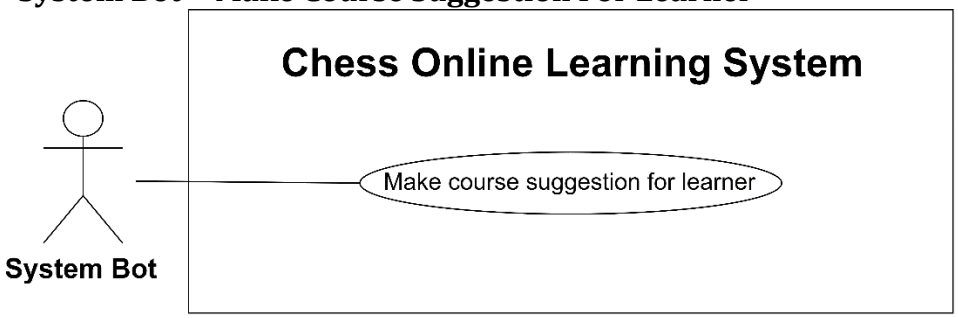


Figure 25: <System Bot> Make Course Suggestion For Learner

USE CASE – UC_SB.03			
Use Case No.	UC_SB.03	**Use Case Version**	2.0
Use Case Name	Make Course Suggestion For Learner		
Author	Phạm Hoàng Tuyết Ngân		
Date	09/09/2019	**Priority**	Normal

Actor:
- System Bot

Summary:
- This use case allows the system to suggest course for learners.

Goal:
- System can suggest appropriate courses for learners.

Triggers:
- Make course suggestion command is sent.

Preconditions: N/A
Postconditions:
- Success: System calculates and synthetic suggestion courses for each learner.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	System calculates course suggestion for learners periodically.	Suggestion course list for each learner.

Alternative Scenario: N/A
Exceptions: N/A
Relationships: N/A
Business Rules:
- Suggestion courses are based on the learning history of learners.
- System finds new courses which are same with learner courses history list.
- System finds new courses by similar learners.

Table 27: Make Course Suggestion For Learner specification

2.3.5 <Instructor> Overview Use Case

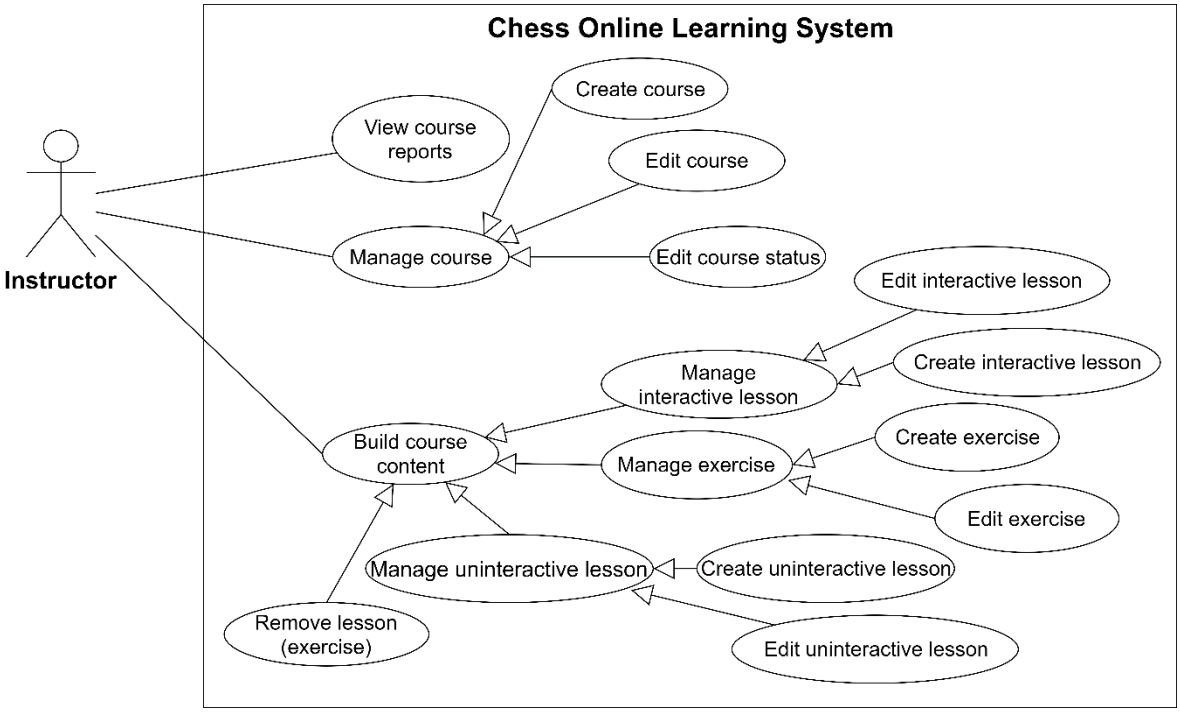


Figure 26: <Instructor> Overview Use Case

2.3.5.1 <Instructor> Create Course

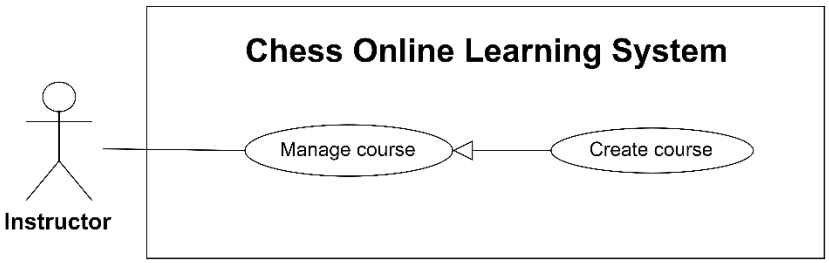


Figure 27: <Instructor> Create Course

USE CASE – UC_IN.01			
Use Case No.	UC_IN.01	**Use Case Version**	2.0
Use Case Name	Create Course		
Author	Phạm Hoàng Tuyết Ngân		
Date	20/07/2019	**Priority**	High

Actor:
- Instructor

Summary:
- This use case allows the instructor to create a course.

Goal:
- New course is created.

Triggers:
- Instructor sends the create course command.

Preconditions:
- User signed in as an instructor successfully.

Postconditions:

- Success: New course is created.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Instructor navigates to Create course page.	System goes to create a course page and shows the form: • Tên khóa học: course name text field • Mô tả: course descrption text area • Yêu cầu: skill level selection, default value is "Mới bắt đầu" • Danh mục: category selection • Ảnh khóa học: course image file field, default is system image • Hoàn tất: submit button • Xóa toàn bộ: reset button System required information: • Tên khóa học: min length 6, max length 255 • Mô tả: min length 6
2	Instructor inputs information.	[Exception 1]
3	Instructor clicks on "Hoàn tất" button. [Alternative 1]	System shows the message "Tạo khóa học thành công" and create course. System redirects to course details page with empty content.

Alternative Scenario:

Step	Actor Action	System Response
1	The instructor click "Xóa toàn bộ" button.	The create course form is reset.

Exceptions:

Step	Cause	System Response
1	Instructor inputs invalid information.	System shows error message match with fields: • Empty name: "Tên khóa học không được bỏ trống" • Name is less than 6 characters: "Tên khóa học không được ít hơn 6 kí tự" • Name is more than 225 characters: "Tên khóa học không được nhiều hơn 255 kí tự" • Empty description: "Mô tả khóa học không được bỏ trống"

		• Description is less than 6 characters: "Mô tả khóa học không được ít hơn 6 kí tự" • Wrong image format: "Hình ảnh không đúng định dạng"

Relationships: N/A
Business Rules:
- Instructor can create courses.
- A new course is added in the database with status is Drafting and empty learning content.

Table 28: Create Course specification

2.3.5.2 <Instructor> Edit Course

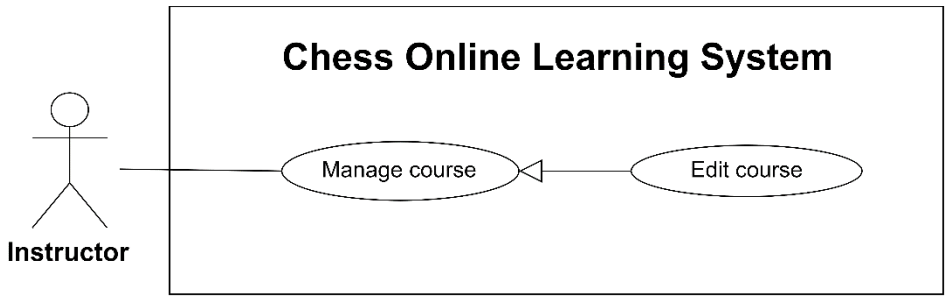


Figure 28: <Instructor> Edit course

USE CASE – UC_IN.02			
Use Case No.	UC_IN.02	**Use Case Version**	2.0
Use Case Name	Edit Course		
Author	Phạm Hoàng Tuyết Ngân		
Date	20/07/2019	**Priority**	High

Actor:
- Instructor

Summary:
- This use case allows the instructor to edit their course information.

Goal:
- Instructor can edit specific course information.

Triggers:
- Instructor sends the edit course command.

Preconditions:
- User signed in as an instructor successfully.
- The instructor must have at least 1 course.

Postconditions:
- Success: Course information is updated.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Instructor navigates to edit a specific course on the course list.	System displays the course information form: • Tên khóa học: course name text field

		• Mô tả: course descrption text area • Yêu cầu: skill level selection, default value is "Mới bắt đầu" • Danh mục: category selection • Ảnh khóa học: course image file field, default is system image • Hoàn tất: submit button • Xóa toàn bộ: reset button System required information: • Tên khóa học: min length 6, max length 255 • Mô tả: min length 6
2	Instructor inputs information.	[Exception 1]
3	Instructor clicks on "Lưu" button. [Alternative 1]	System updates course information and display message "Chỉnh sửa khóa học thành công". A notification occurs to users.

Alternative Scenario:

Step	Actor Action	System Response
1	The instructor clicks on "Hủy" button	System returns to the course details page without changing.

Exceptions:

Step	Cause	System Response
1	Instructor inputs invalid information.	System shows error message match with fields: • Empty name: "Tên khóa học không được bỏ trống" • Name is less than 6 characters: "Tên khóa học không được ít hơn 6 kí tự" • Name is more than 225 characters: "Tên khóa học không được nhiều hơn 255 kí tự" • Empty description: "Mô tả khóa học không được bỏ trống" • Description is less than 6 characters: "Mô tả khóa học không được ít hơn 6 kí tự" • Wrong image format: "Hình ảnh không đúng định dạng"

Relationships: N/A

Business Rules:

- Instructor can only edit course information.

• If the course was in Published status, a notification will be sent to admin and enrolled learner.

Table 29: Edit Course specification

2.3.5.3 <Instructor> Edit Course Status

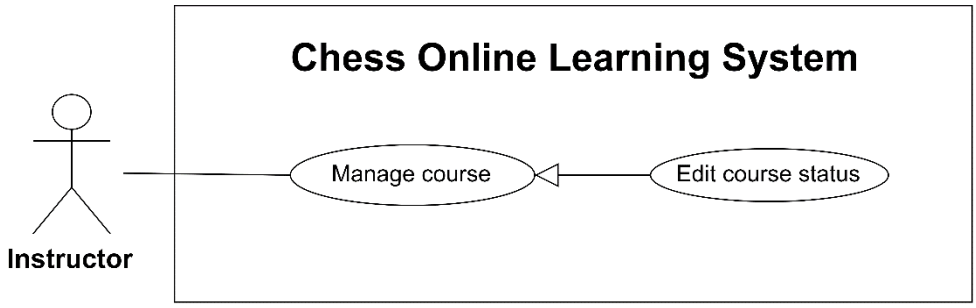


Figure 29: <Instructor> Edit Course Status

USE CASE – UC_IN.03			
Use Case No.	UC_IN.03	**Use Case Version**	2.0
Use Case Name	Edit Course Status		
Author	Phạm Hoàng Tuyết Ngân		
Date	20/07/2019	**Priority**	High

Actor:
- Instructor

Summary:
- This use case allows instructors to change the status of their courses.

Goal:
- Instructors can change their course status.

Triggers:
- Instructor sends the edit course status command.

Preconditions:
- User signed in as an instructor successfully.
- The instructor must have at least 1 course.

Postconditions:
- Success: Course status is changed.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Instructor clicks on "Setting" button in a specific course.	System displays actions for instructor: • Submit course: for drafting course • Unpublish course: for published course • Restore course: for rejected course • Resist course: for waiting course

2	Instructor request to change course status.	System displays confirm dialogue with confirming message.
3	Instuctor clicks on "Xác nhận". [Alternative 1]	Course status is changed and the system occurs to users.

Alternative Scenario:

Step	Actor Action	System Response
1	Instructor clicks on "Đóng" button.	System closes the dialogue without changing.

Exceptions: N/A
Relationships: N/A
Business Rules:

- Instructor can unpublish their Published courses.
- Instructor can submit their Drafting courses if lesson list is not empty.
- Instructor can restore their Rejected courses.
- Instructor can resist their Waiting courses.
- After action successfully, the course status is transformed.
- System will send a notification to admin and enrolled learners if the course is unpublished.
- System will send a notification to admin if the course is submitted.

Table 30: Edit Course Status specification

2.3.5.4 <Instructor> View Course Reports

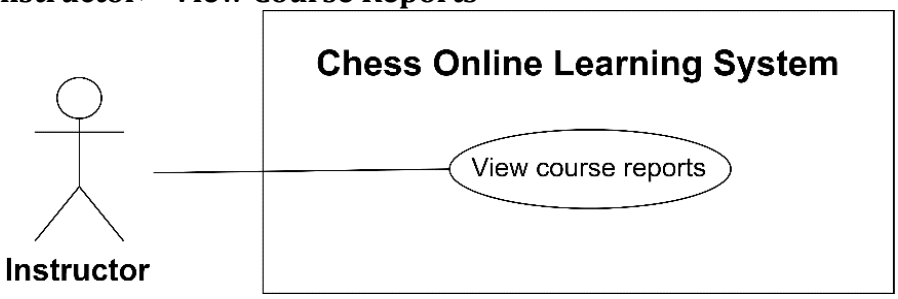


Figure 30: <Instructor> View Course Reports

USE CASE – UC_IN.04			
Use Case No.	UC_IN.04	**Use Case Version**	2.0
Use Case Name	View Course Reports		
Author	Đặng Văn Hoàng		
Date	20/07/2019	**Priority**	Normal

Actor:
- Instructor

Summary:
- This use case allows the instructor to view the course registration statistics.

Goal:
- Read course statistics chart.

Triggers:
- Instructor sends the view course reports command.

Preconditions:
- User signed in as an instructor successfully.
- The instructor must have at least 1 course.

Postconditions:
- Success: System displays statistics charts of the course.
- Failed: System displays error messages.

Main Success Scenario:

Step	Actor Action	System Response

1	Instructor navigates to report page.	System shows:
		• Registration chart by month: line chart
		  ○ Year Search: text field
		• Number and result of learners by course: table
		  ○ Course name search: text field
		  ○ Number of rows: selectbox
		[Exception 1]

Alternative Scenario: N/A

Exceptions:

Step	Cause	System Response
1	Instructor does not have any course.	Error message: "Không có gì để thống kê".

Relationships: N/A

Business Rules:
- System can calculate the number of enrolled learners for each instructor by month.
- Instructor can search learner statistics by year.
- System can calculate the number of learners for each course by learning progress: Finished, Studying, Not Finished.

Table 31: View Course Reports specification

2.3.5.5 <Instructor> Create Interactive Lesson

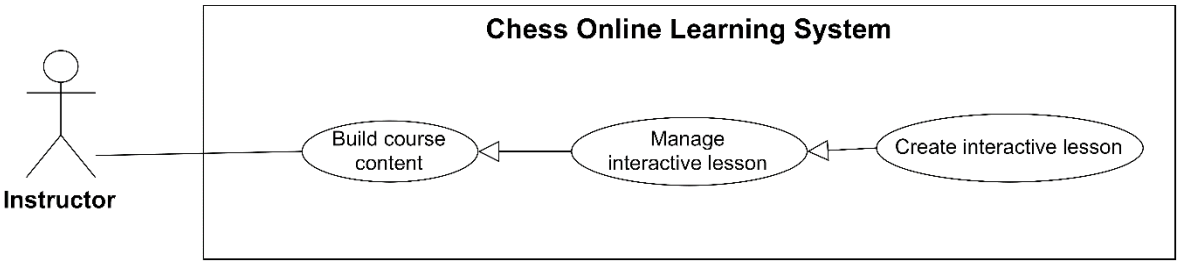


Figure 31: <Instructor> Create Interactive Lesson

USE CASE – UC_IN.05			
Use Case No.	UC_IN.05	**Use Case Version**	2.0
Use Case Name	Create Interactive Lesson		
Author	Phạm Hoàng Tuyết Ngân		
Date	20/07/2019	**Priority**	High

Actor:
- Instructor

Summary:
- This use case allows the instructor to create an interactive lesson.

Goal:
- New interactive lesson is created successfully.

Triggers:
- Instructor sends the create interactive lesson command.

Preconditions:
- User signed in as an instructor successfully.
- Instructor must have at least 1 course.

Postconditions:
- Success: New lesson is created to course.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Instructor requests to create an interactive lesson.	System displays a screen to add interactive lessons: • Tên bài học: text field • Mô tả: text area • Nội dung: text area • Nước đi: move content-box • Xóa toàn bộ: button • Chessboard • FEN Editing board button • Xem trước: button (disable, enable when information is validated) • Trở về: button (disable, enable when user the is in preview mode) • Close button System required information:

		• Tên bài học: min length 6, max length 255 • Mô tả: min length 6 • The move list is not empty
2	Instructor input lesson name and description.	[Exception 1]
3	Instructor creates lesson content by performing move on the chessboard. [Alternative 1]	Move is auto-added in move content box.
4	Instructor can add content for each move in "Nội dung" field.	Content is added for the move.
5	Instructor clicks "Xem trước" button [Alternative 2] [Alternative 3]	System displays lesson content as live mode: • Lưu: button • Trở về: button
6	Instructor clicks "Lưu" button. [Alternative 2]	System displays a message "Tạo trận đấu thành công" and create new interactive lesson.

Alternative Scenario:

Step	Actor Action	System Response
1a	Instructor clicks FEN editing board button.	System shows form "Tạo thế cờ" • Chessboard • Lưu thế cờ: button • Đóng: button.
1b	Instructor drags chess pieces to the chessboard and clicks the save button.	System shows a new chess position on the chessboard and all existing moves is removed.
2	Instructor clicks the close button.	System closes create lesson form and return to previous page.
3	Instructor clicks the remove all button.	System removes all the information were set up.

Exceptions:

Step	Cause	System Response
1	Instructor inputs invalid information in the text fields.	System shows error message match with fields: • Empty name: "Tên bài học không được bỏ trống" • Name is less than 6 characters: "Tên bài học không được ít hơn 6 kí tự" • Name is more than 225 characters: "Tên bài học không được nhiều hơn 255 kí tự" • Empty description: "Mô tả bài học không được bỏ trống"

		• Description is less than 6 characters: "Mô tả bài học không được ít hơn 6 kí tự"

Relationships: N/A
Business Rules:
- Instructor creates a new interactive lesson to build course content.
- Interaction lesson consists of many different chess moves and moves analyse, the instructor can freely set the chessboard.
- A course can have many interactive lessons.

Table 32: Create Interactive Lesson specification

2.3.5.6 <Instructor> Edit Interactive Lesson

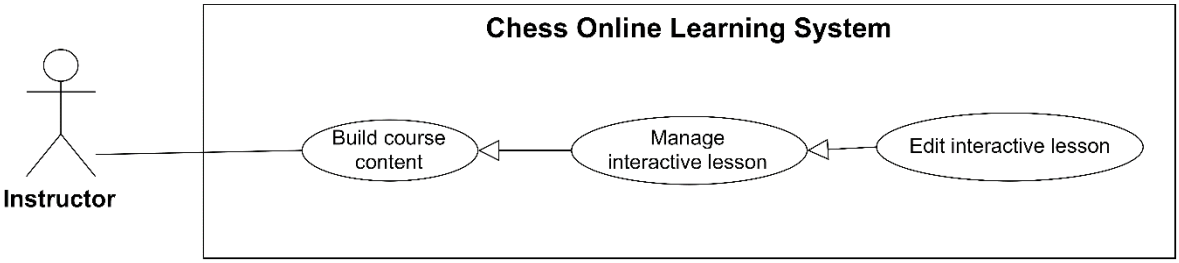


Figure 32: <Instructor> Edit Interactive Lesson

USE CASE – UC_IN.06			
Use Case No.	UC_IN.06	**Use Case Version**	2.0
Use Case Name	Edit Interactive Lesson		
Author	Phạm Hoàng Tuyết Ngân		
Date	20/07/2019	**Priority**	Normal

Actor:
- Instructor

Summary:
- This use case allows the instructor to edit an interactive lesson.

Goal:
- Instructor can edit content of an interactive lesson.

Triggers:
- Instructor send the edit interactive lesson command.

Preconditions:
- User signed in as an instructor successfully.
- The instructor must have at least 1 interactive lesson.

Postconditions:
- Success: System updates interactive lesson content successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Instructor navigates to edit the specific interactive lesson.	System loads existing content and displays a form to edit interactive lessons: • Tên bài học: text field • Mô tả: text area

		• Nội dung: text area • Nước đi: move content-box • Xóa toàn bộ: button • Chessboard • FEN Editing board button • Xem trước: button (disable, enable when information is validated) • Trở về: button (disable, enable when using the preview mode) • Close button System required information: • Tên bài học: min length 6, max length 255 • Mô tả: min length 6 • The move list is not empty
2	Instructor edit lesson name and description.	[Exception 1]
3	Instructor edit lesson content by performing moves on chessboard. [Alternative 1]	Move is auto-added in move content-box
4	Instructor can edit content for each move.	Move content is changed.
5	Instructor clicks "Xem trước" button [Alternative 2] [Alternative 3]	System displays lesson content as live mode: • Lưu: button • Trở về: button.
6	Instructor clicks "Lưu" button. [Alternative 2]	System displays a message "Sửa trận đấu thành công" and edit interactive lessons content.

Alternative Scenario:

Step	Actor Action	System Response
1a	Instructor clicks FEN editing board button.	System shows form "Tạo thế cờ" • Chessboard • Lưu thế cờ: button • Đóng: button.
1b	Instructor drags chess pieces to the chessboard and clicks the save button.	System shows a new chess position on the chessboard and all existing moves is removed.
2	Instructor clicks "Đóng" button	System closes the lesson form and return to the previous page.
3	Instructor clicks "Xóa toàn bộ"	System removes all the information were set up.

Exceptions:

Step	Cause	System Response
1	Instructor inputs invalid information in the text fields.	System shows error message match with fields:

		• Empty name: "Tên bài học không được bỏ trống" • Name is less than 6 characters: "Tên bài học không được ít hơn 6 kí tự" • Name is more than 225 characters: "Tên bài học không được nhiều hơn 255 kí tự" • Empty description: "Mô tả bài học không được bỏ trống" • Description is less than 6 characters: "Mô tả bài học không được ít hơn 6 kí tự"

Relationships: N/A
Business Rules:
- The instructor can edit interactive lessons in their course.
- When editing the interactive lesson in the published course, the lessons will also be edited accordingly on the learner page.
- System will send a notification to admin and enrolled learner.

Table 33: Edit Interactive Lesson specification

2.3.5.7 <Instructor> Create Uninteractive Lesson

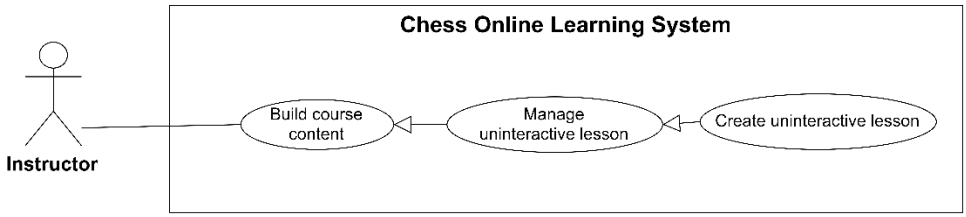


Figure 33: <Instructor> Create Uninteractive Lesson

USE CASE – UC_IN.07			
Use Case No.	UC_IN.07	**Use Case Version**	2.0
Use Case Name	Create Uninteractive Lesson		
Author	Đặng Văn Hoàng		
Date	20/07/2019	**Priority**	High

Actor:
- Instructor

Summary:
- This use case allows the instructor to create an uninteractive lesson.

Goal:
- Instructor can create an uinteractive lesson for the course.

Triggers:
- Instructor sends the create uninteractive lesson commands.

Preconditions:
- User signed in as an instructor successfully.
- The instructor must have at least 1 course.

Postconditions:
- Success: An uninteractive lesson is added to the course.

- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Instructor navigates to creates an unteractive lesson	System displays a form to create the uninteractive lesson: • Tên bài học: text field • Mô tả: text area • Nội dung bài học: WYSIWG Editor • Thêm: button • Close button System required information: • Tên bài học: min length 6, max length 255 • Mô tả: min length 6 • Nội dung: min length 6
2	Instructor inputs information.	[Exception 1]
3	Instructor clicks on "Thêm" button. [Alternative 1]	System create uninteractive lessons and displays message "Tạo bài đọc học thành công".

Alternative Scenario:

Step	Actor Action	System Response
1	Instructor clicks on the close icon.	System closes the form without changing.

Exceptions:

Step	Cause	System Response
1	Instructor inputs invalid information or not fill in WYSIWYG Editor.	System shows error message match with fields: • Empty name: "Tên bài học không được bỏ trống" • Name is less than 6 characters: "Tên bài học không được ít hơn 6 kí tự" • Name is more than 225 characters: "Tên bài học không được nhiều hơn 255 kí tự" • Empty description: "Mô tả bài học không được bỏ trống" • Description is less than 6 characters: "Mô tả bài học không được ít hơn 6 kí tự" • Empty content: "Nội dung bài học không được bỏ trống" • Content is less than 6 characters: "Nội dung bài học không được ít hơn 6 kí tự"

Relationships: N/A
Business Rules:
- Instructor can create uninteractive lessons to build course content.
- Uninteractive lesson is a theoretical learning format including images, videos, document files.
- A course can have many uninteractive lessons.

Table 34: Create Uniteractive Lesson specification

2.3.5.8 <Instructor> Edit Uninteractive Lesson

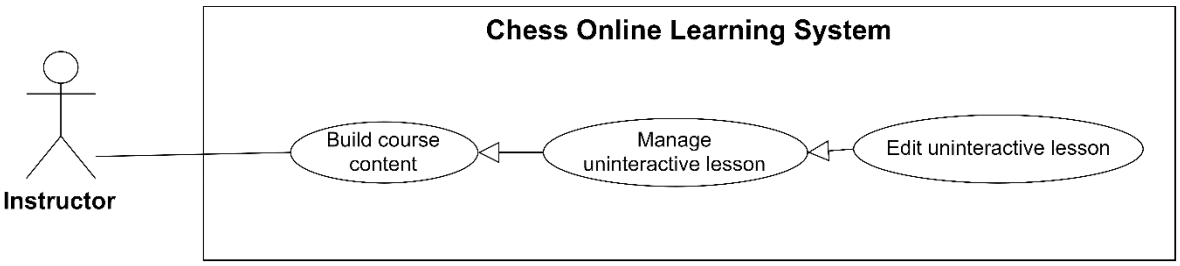


Figure 34: <Instructor> Edit Uninteractive Lesson

USE CASE – UC_IN.08			
Use Case No.	UC_IN.08	**Use Case Version**	2.0
Use Case Name	Edit Uninteractive Lesson		
Author	Đặng Văn Hoàng		
Date	20/07/2019	**Priority**	High

Actor:
- Instructor

Summary:
- This use case allows the instructor to edit uninteractive lesson.

Goal:
- Instructor can edit uinteractive lesson content.

Triggers:
- Instructor sends the edit uninteractive lesson command.

Preconditions:
- User signed in as an instructor successfully.
- The instructor must have at least 1 uninteractive lesson.

Postconditions:
- Success: System update content for uninteractive lesson successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Instructor navigates to edit a specific uninteractive lesson.	System load existing content and displays on a form: • Tên bài học: text field • Mô tả: text area • Nội dung bài học: WYSIWG Editor • Thêm: button • Close button

		System required information: • Tên bài học: min length 6, max length 255 • Mô tả: min length 6 • Nội dung: min length 6
2	Instructor edit lesson information.	[Exception 1]
3	Instructor clicks on "Thêm" button. [Alternative 1]	System updates uninteractive lessons and shows a message "Sửa bài học thành công".

Alternative Scenario:

Step	Actor Action	System Response
1	Instructor clicks on the close icon.	System closes the form without changing.

Exceptions:

Step	Cause	System Response
1	Instructor inputs invalid information or not fill in WYSIWYG Editor.	System shows error message match with fields: • Empty name: "Tên bài học không được bỏ trống" • Name is less than 6 characters: "Tên bài học không được ít hơn 6 kí tự" • Name is more than 225 characters: "Tên bài học không được nhiều hơn 255 kí tự" • Empty description: "Mô tả bài học không được bỏ trống" • Description is less than 6 characters: "Mô tả bài học không được ít hơn 6 kí tự" • Empty content: "Nội dung bài học không được bỏ trống" • Content is less than 6 characters: "Nội dung bài học không được ít hơn 6 kí tự"

Relationships: N/A

Business Rules:

- The instructor can edit uninteractive lessons content.
- When editing the uninteractive lesson in the public course, the lessons will also be edited accordingly on the learner page.
- System will send a notification to admin and enrolled learner.

Table 35: <Instructor> Edit Uninteractive Lesson sepecification

2.3.5.9 <Instructor> Create Exercise

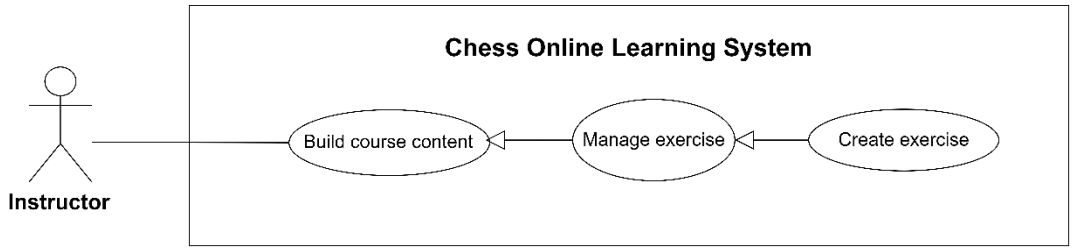


Figure 35: <Instructor> Create Exercise

USE CASE – UC_IN.09			
Use Case No.	UC_IN.09	**Use Case Version**	2.0
Use Case Name	Create Exercise		
Author	Phạm Hoàng Tuyết Ngân		
Date	20/07/2019	**Priority**	High

Actor:
- Instructor

Summary:
- This use case allows the instructor to create an exercise.

Goal:
- Instructor can create a new exercise.

Triggers:
- Instructor sends the create exercise command.

Preconditions:
- User signed in as an instructor successfully.
- The instructor must have at least 1 course.

Postconditions:
- Success: An exercise is added to the course.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response

1	Instructor navigates to create a new exercise.	System displays a form to create an exercise:
		• Tên bài tập: text field
		• Câu hỏi: text field
		• Mô tả: text area
		• Three of the answer:
		  ○ Nước đi: move content-box
		  ○ Thực hiện đúng: text field
		  ○ Thực hiện sai: text field
		• Xóa toàn bộ: button
		• Chessboard
		• FEN Editing board button
		• Xem trước: button (disable/enable when information is valid)
		• Trở về: button (disable/enable when the user is in preview mode)

		• Close button • Dùng chế độ đánh tự động: checkbox System required information: • Tên bài tập: min length 6, max length 255 • Câu hỏi: min length 6 • Mô tả: min length 6 • Answer List is not empty or enable chess bot mode
2	Instructor inputs exercise name, question and description.	[Exception 1]
3	Instructor adds lesson content by performing a move. [Alternative 1] [Alternative 2]	Move is auto-added to move content-box.
4	Instructor can fill "Thực hiện đúng" and "Thực hiện sai" content for each move.	System saves the right response and wrong response for the move.
5	Instructor clicks "Xem trước" button. [Alternative 3] [Alternative 4]	System shows the new exercise in live mode: • Lưu: button • Trở về: button.
6	Instructor clicks "Lưu" button. [Alternative 3]	System creates a new exercise and shows a message "Thêm bài tập thành công".

Alternative Scenario:

Step	Actor Action	System Response
1a	Instructor clicks FEN editing board button.	System shows form "Tạo thế cờ" • Chessboard • Lưu thế cờ: button Đóng: button.
1b	Instructor drags chess pieces to the chessboard and clicks the save button.	System shows a new chess position on the chessboard and all existing moves is removed.
2	Instructor chooses bot answer.	Move answer form is disabled and the answer is noted in a bot answer.
3	Instructor clicks on the close button.	System closes the for without changing.
4	Instructor clicks on the remove all button.	System removes all the instructor information which were set up.

Exceptions:

Step	Cause	System Response
1	Instructor inputs invalid information in the text fields.	System shows error message match with fields: • Empty name: "Tên bài tập không được bỏ trống"

		• Name is less than 6 characters: "Tên bài tập không được ít hơn 6 kí tự" • Name is more than 225 characters: "Tên bài tập không được nhiều hơn 255 kí tự" • Empty question: "Câu hỏi không được bỏ trống" • Question is less than 6 characters: "Câu hỏi bài học không được ít hơn 6 kí tự" • Empty is description: "Mô tả bài tập không được bỏ trống" • Description is less than 6 characters: "Mô tả bài tập không được ít hơn 6 kí tự"

Relationships: N/A
Business Rules:
- Instructor can create exercises to build course content.
- A course can have many exercises.
- Exercise is a lesson, which provides a chess puzzle and learner must perform the right move to solve the problem.
- Exercise contains 2 types of answer: using Bot answer and Manual answer.
- There are as much as possible 3 answers in Manual answer.
- Manual answer support right response and wrong response for each move.

Table 36: <Instructor> Create Exercise specification

2.3.5.10 <Instructor> Edit Exercise

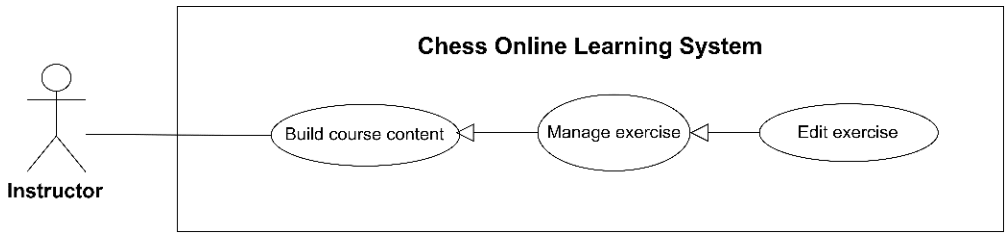


Figure 36: <Instructor> Edit Exercise

USE CASE – UC_IN.10			
Use Case No.	UC_IN.10	**Use Case Version**	2.0
Use Case Name	Edit Exercise		
Author	Phạm Hoàng Tuyết Ngân		
Date	20/07/2019	**Priority**	High

Actor:
- Instructor

Summary:
- This use case allows the instructor to edit exercises content.

Goal:
- Instructor can edit an exercise.

Triggers:
- Instructor sends the edit exercise commands.

Preconditions:
- User signed in as an instructor successfully.
- The instructor must have at least 1 exercises.

Postconditions:
- Success: Exercise content is edited successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response

1	Instructor navigates to edit a specific exercise.	System load existing exercise and displays to form:
		• Tên bài tập: text field
		• Câu hỏi: text field
		• Mô tả: text area
		• Three of the answer:
		  ○ Nước đi: move content-box
		  ○ Thực hiện đúng: text field
		  ○ Thực hiện sai: text field
		• Xóa toàn bộ: button
		• Chessboard
		• FEN Editing board button
		• Xem trước: button (disable/enable when information is valid)
		• Trở về: button (disable/enable when the user is in preview mode)
		• Close button
		• Dùng chế độ đánh tự động: checkbox
		System required information:
		• Tên bài tập: min length 6, max length 255
		• Câu hỏi: min length 6
		• Mô tả: min length 6
		Answer List is not empty or enable chess bot mode

2	Instructor edit lesson name, question and description.	[Exception 1]
	Instructor edit exercise content by perform moves. [Alternative 1] [Alternative 2]	Moves is auto-edited to move content-box.
3	Instructor can edit the right response and wrong response for each move.	Move right response and wrong response is changed.
4	Instructor clicks “Xem trước” button.	System displays new exercise in live mode:

	[Alternative 3] [Alternative 4]	• Lưu: button • Trở về: button.
5	Instructor clicks "Lưu" button. [Alternative 3]	System updated exercise content and displays message "Sửa bài tập thành công". [Exception 1]

Alternative Scenario:

Step	Actor Action	System Response
1a	Instructor clicks FEN editing board button.	System shows form "Tạo thế cờ" • Chessboard • Lưu thế cờ: button Đóng: button.
1b	Instructor drags chess pieces to the chessboard and clicks the save button.	System shows a new chess position on the chessboard and all existing moves is removed.
2	Instructor chooses bot answer.	Move answer form is disabled and the answer is noted to bot answer.
3	Instructor clicks on the close button.	Close form without add new exercise.
4	Instructor clicks on the remove all button.	System removes all which the instructor was set up.

Exceptions:

Step	Cause	System Response
1	Instructor inputs invalid information.	System shows error message match with fields: • Empty name: "Tên bài tập không được bỏ trống" • Name is less than 6 characters: "Tên bài tập không được ít hơn 6 kí tự" • Name is more than 225 characters: "Tên bài tập không được nhiều hơn 255 kí tự" • Empty question: "Câu hỏi không được bỏ trống" • Question is less than 6 characters: "Câu hỏi bài học không được ít hơn 6 kí tự" • Empty is description: "Mô tả bài tập không được bỏ trống" • Description is less than 6 characters: "Mô tả bài tập không được ít hơn 6 kí tự"

Relationships: N/A
Business Rules:

• The instructor can edit exercise content. • When editing the exercise in the public course, the exercise will also be edited accordingly on the learner page. • System will send a notification to admin and enrolled learner.

Table 37: Edit Exercise specification

2.3.5.11 <Instructor> Remove Lesson (Exercise)

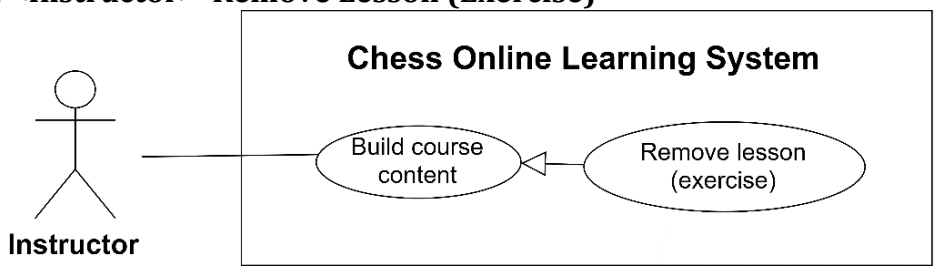


Figure 37: <Instructor> Remove Lesson (Exercise)

USE CASE – UC_IN.11			
Use Case No.	UC_IN.11	**Use Case Version**	2.0
Use Case Name	Remove Lesson (Exercise)		
Author	Phạm Hoàng Tuyết Ngân		
Date	20/07/2019	**Priority**	Normal

Actor:
- Instructor

Summary:
- This use case allows the instructor can remove a lesson or an exercise.

Goal:
- Instructor can remove the lesson (exercise) in course.

Triggers:
- Instructor sends the remove lesson (exercise) command.

Preconditions:
- User signed in as an instructor successfully.
- Instructor must have at least 1 lessons (exercise) in course.

Postconditions:
- Success: A lesson (exercise) is removed successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Instructor requests to remove a specific lesson (exercise)	System displays confirmation dialogue: • Bạn có muốn xóa bài học này: text • Xác nhận: button • Đóng: button
2	Instructor clicks on "Xác nhận" button [Alternation 1]	System remove lesson (exercise) and displays a message "Xóa bài học thành công".

Alternative Scenario:

Step	Actor Action	System Response

1	Instructor clicks on "Đóng" button	System closes confirmation dialogue without changing.

Exceptions: N/A
Relationships: N/A
Business Rules:

- Instructor can remove lessons (exercise) in a specific course.
- When removing lessons (exercises) in the public course, system will send a notification to admin and enrolled learner.

Table 38: Remove Lesson (Exercise) specification

2.3.6 <Admin> Overview Use Case

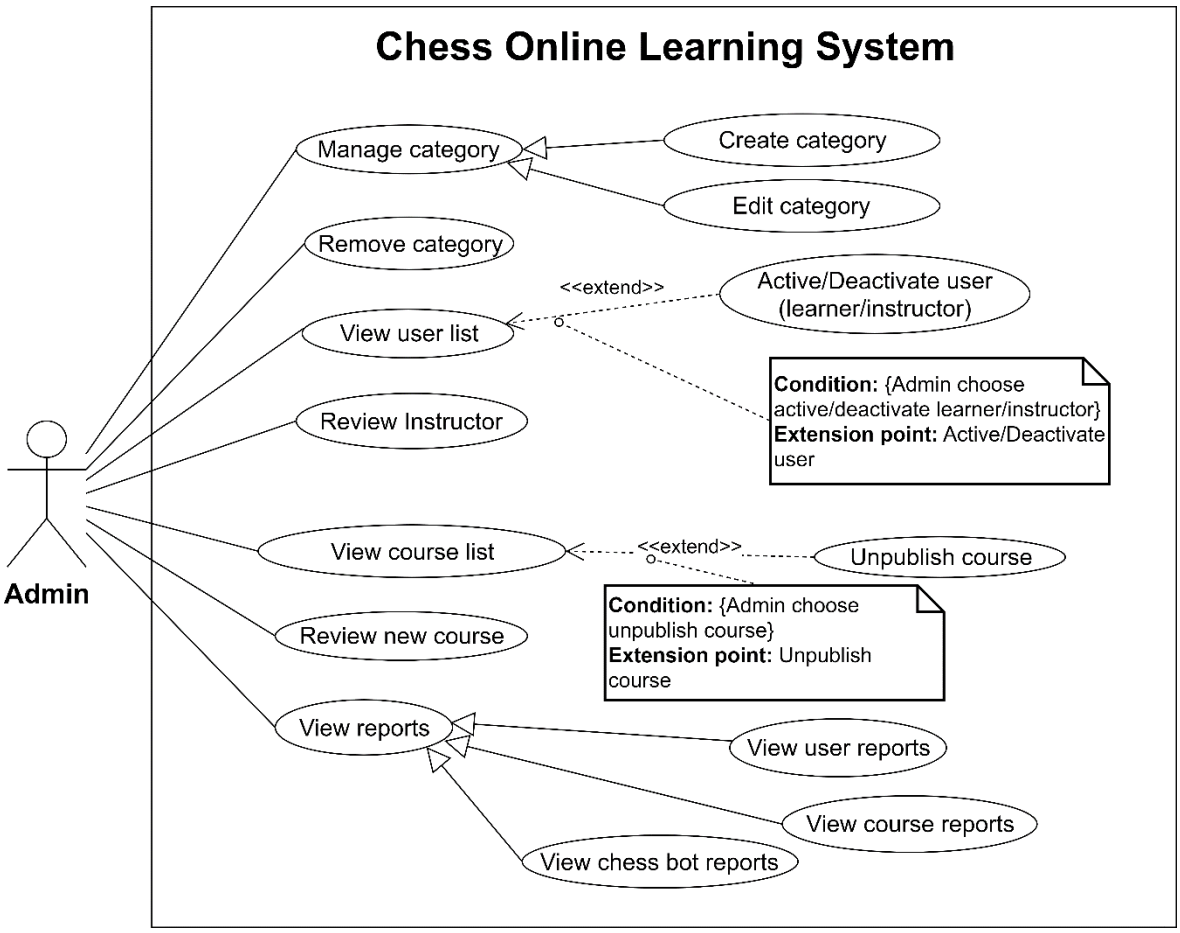


Figure 38: <Admin> Overview Use Case

2.3.6.1 <Admin> Create Category

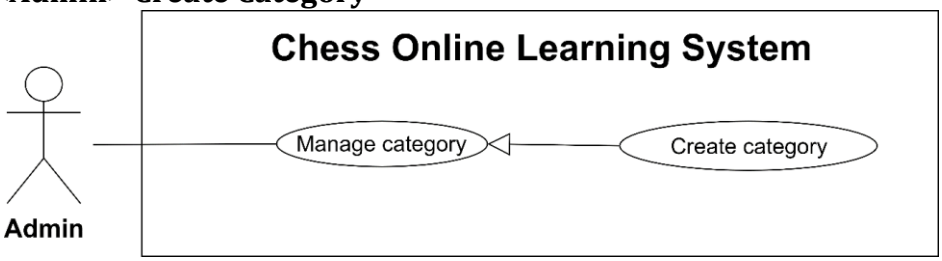


Figure 39: <Admin> Create Category

USE CASE – UC_AD.01

Use Case No.	UC_AD.01	Use Case Version	2.0
Use Case Name	Create Category		
Author	Đặng Văn Hoàng		
Date	20/7/2019	Priority	Normal

Actor:
- Admin

Summary:
- This use case allows the admin to create a category.

Goal:
- A new category is added to system.

Triggers:
- Admin sends the create category command.

Preconditions:
- User signed in as an admin successfully.

Postconditions:
- Success: A category is added to the system successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin requests to create a new category.	System shows create category form: • Danh mục: text field • Đồng ý: button • Hủy: button System required information: • Danh mục: min length 6, max length 255
2	Admin inputs text field.	[Exception 1]
3	Admin clicks on "Đồng ý" button [Alternative 1]	System shows the message "Tạo danh mục thành công" and a new category is added.

Alternative Scenario:

No	Actor Action	System Response
1	Admin clicks on "Hủy" button	System returns to the category management page without changing.

Exceptions:

No	Cause	System Response
1	Admin inputs invalid information.	System shows error message match with fields: • Empty name: "Tên danh mục không được bỏ trống"

		• Name is less than 6 characters: "Tên danh mục không được ít hơn 6 kí tự" • Name is more than 225 characters: "Tên danh mục không được nhiều hơn 225 kí tự"

Relationships: N/A
Business Rules:
- System category can be created by admins.

Table 39: Create Category specification

2.3.6.2 <Admin> Edit Category

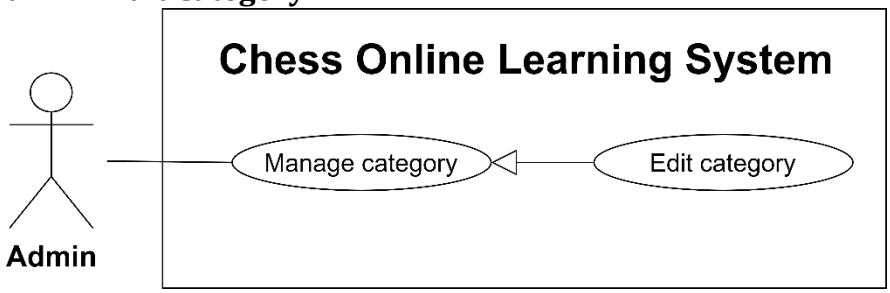


Figure 40: <Admin> Edit Category

USE CASE – UC_AD.02			
Use Case No.	UC_AD.02	**Use Case Version**	1.0
Use Case Name	Edit Category		
Author	Đặng Văn Hoàng		
Date	20/7/2019	**Priority**	Normal

Actor:
- Admin

Summary:
- This use case allows the admin to edit category.

Goal:
- Edit name of a specific category.

Triggers:
- Admin sends the edit category command.

Preconditions:
- User signed in as an Admin successfully.
- System must have at least 1 category.

Postconditions:
- Success: a category name is edited successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin requests to edit a specific category.	System shows create category form:

		• Danh mục: text field • Đồng ý: button • Hủy: button System required information: • Danh mục: min length 6, max length 255
2	Admin inputs text field.	[Exception 1]
3	Admin clicks on "Đồng ý" button [Alternative 1]	System shows the message "Chỉnh sửa danh mục thành công" and category name is edited.

Alternative Scenario:

No	Actor Action	System Response
1	Admin clicks on "Hủy" button	System returns to the category management page without changing.

Exceptions:

No	Cause	System Response
1	Admin inputs invalid information.	System shows error message match with fields.

Relationships: N/A

Business Rules:

- Category name can be edited by admins.
- If the category name is edited, the category names of the courses will be changed accordingly.

Table 40: Edit Category specification

2.3.6.3 <Admin> Remove Category

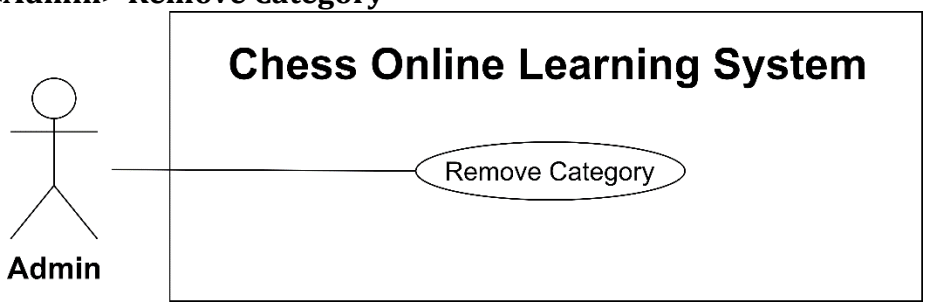


Figure 41: <Admin> Remove Category

USE CASE – UC_AD.03			
Use Case No.	UC_AD.03	**Use Case Version**	2.0
Use Case Name	Remove Category		
Author	Đặng Văn Hoàng		
Date	20/7/2019	**Priority**	Normal

Actor:
- Admin

Summary:
- This use case allows the admin to remove system categories.

Goal:
- A system category is removed.

Triggers:
- Admin sends the remove category command.

Preconditions:
- User signed in as an admin successfully.
- System must have at least 1 category.

Postconditions:
- Success: Category is removed successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin requests to remove a specific category.	System shows remove category confirmation: • Bạn có chắc muốn xóa danh mục: text • Đồng ý: button • Hủy: button
2	Admin click on "Đồng ý" button. [Alternative 1]	System shows the message "Xóa danh mục thành công" and category is removed.

Alternative Scenario:

No	Actor Action	System Response
1	Admin clicks on "Hủy" button	System returns to the category management page without changing.

Exceptions: N/A
Relationships: N/A
Business Rules:
- If the category name is removed, the relationship between courses and this category is removed.

Table 41: Remove Category specification

2.3.6.4 <Admin> View User Reports

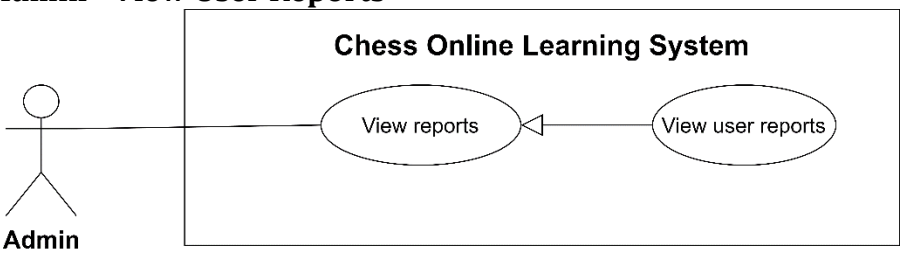


Figure 42: <Admin> View User Reports

USE CASE – UC_AD.04			
Use Case No.	UC_AD.04	**Use Case Version**	1.0
Use Case Name	View User Reports		
Author	Đặng Văn Hoàng		
Date	20/7/2019	**Priority**	Normal

Actor:
- Admin

Summary:
- This use case allows admin to view user registration statistic.

Goal:
- Admin can view user registration chart.

Triggers:
- Admin sends the view user reports command.

Preconditions:
- User signed in as an admin successfully.

Postconditions:
- Success: System displays user statistic successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin navigates to view statistics.	System shows user registration charts and text field to search by year: • User Registration by month and provider: Line Chart

Alternative Scenario: N/A
Exceptions: N/A
Relationships: N/A
Business Rules:
- System calculates user registration by the provider for each month.
- Admin can know if the account registration is increasing or decreasing.

Table 42: View User Reports specification

2.3.6.5 <Admin> View Bot Reports

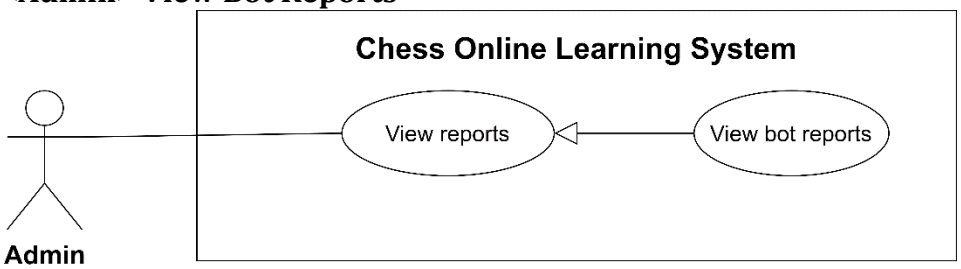


Figure 43: <Admin> View Bot Reports

USE CASE – UC_AD.05			
Use Case No.	UC_AD.05	**Use Case Version**	2.0

Use Case Name	View Bot Reports		
Author	Đặng Văn Hoàng		
Date	20/7/2019	Priority	Normal

Actor:
- Admin

Summary:
- This use case allows admin to view winning rates against bots.

Goal:
- Admin views bot statistics chart.

Triggers:
- Admin sends the view bot reports command.

Preconditions:
- User signed in as an admin successfully.

Postconditions:
- Success: System displays bot statistic chart.
- Failed: System displays error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin navigates to view statistics.	System shows bot statistic and a text field to search year: • Chess game result by month: collum chart • Percentage by level: pie chart

Alternative Scenario:N/A
Exceptions: N/A
Relationships: N/A
Business Rules:
- System stores chess result of learners and makes statistics.
- System calculates result: win, lose, draw or giving up by month.
- System calculates the percentage of games bythe level of all learners.

Table 43: View Bot Reports specification

2.3.6.6 <Admin> View Course Reports

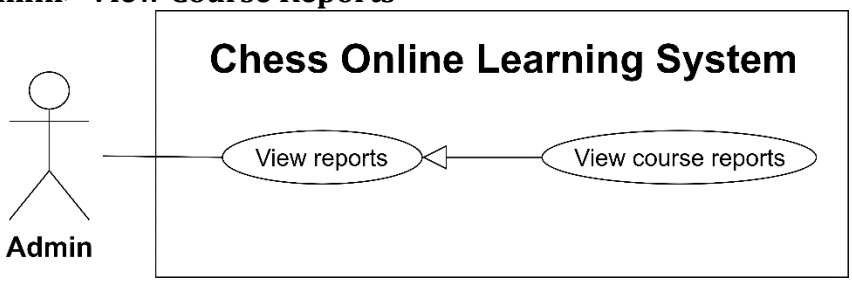


Figure 44: <Admin> View Course Reports

USE CASE – UC_AD.06			
Use Case No.	UC_AD.06	Use Case Version	1.0
Use Case Name	View Course Reports		
Author	Đặng Văn Hoàng		
Date	20/7/2019	Priority	Normal

Actor:
- Admin

Summary:
- This use case allows admin to view course statistic in system.

Goal:
- Admin views course statistics chart.

Triggers:
- Admin sends the view course reports command.

Preconditions:
- User signed in as an admin successfully.

Postconditions:
- Success: System displays course reports.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin navigates to view statistics.	System shows course reports: • Number of public course: column chart • Percentage of course by status: pie chart

Alternative Scenario: N/A
Exceptions: N/A
Relationships: N/A
Business Rules:
- System calculates number of public course.
- System calculates the percentage of courses by their status: Published, Rejected, Drafting, Waiting.
- The system will automatically statistics.

Table 44: View Course Reports specification

2.3.6.7 <Admin> Review New Course

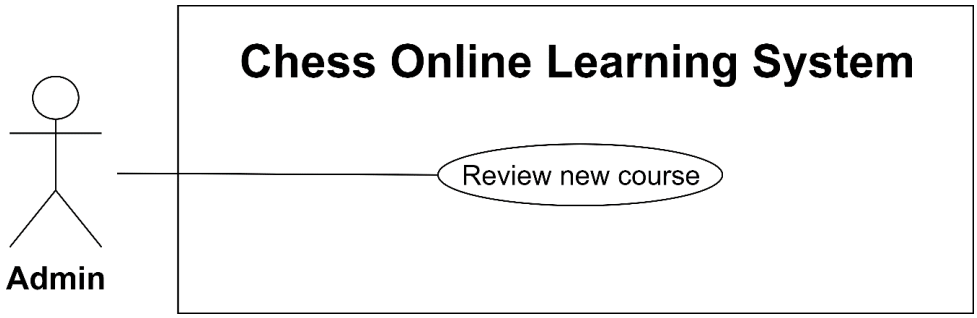


Figure 45: <Admin> Review New Course

USE CASE – UC_AD.07			
Use Case No.	UC_AD.07	**Use Case Version**	2.0
Use Case Name	Review New Course		
Author	Đặng Văn Hoàng		
Date	20/7/2019	**Priority**	Normal

Actor:
- Admin

Summary:
- This use case allows admin review new course. See details of the course: course information and course content.

Goal:
- A course is reviewed before publishing.

Triggers:
- Admin sends the review new course command.

Preconditions:
- User signed in as an admin successfully.

Postconditions:
- Success: A new course is published or rejected.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin navigates to waiting course.	System displays the course details: • Course information: name, description, skill level required, categories. • Course content: the number of lessons, lesson list. • Setting button
2	Admin clicks on the setting button [Alternative 1]	System show dialog "Thay đổi trạng thái khóa học": • Đồng ý: button • Từ chối: button • Đóng: button
3	Admin click on "Đồng ý" button [Alternative 1] [Alternative 2]	System displays a message "Khóa học đã được công khai" and change status course.

Alternative Scenario:

No	Actor Action	System Response
1	Admin clicks on "Đóng" button	System closes dialog without changing in course.
2a	Admin clicks on "Từ chối" button	System displays dialog to enter rejected reason, after Admin confirm reason.
2b	Admin input rejected reson.	Course status is transformed to "Rejected" and system displays a message "Khóa học đã bị từ chối công khai".

Exceptions: N/A
Relationships: N/A
Business Rules:

• When the admin review is completed, the system sends a notifications and confirm email to the instructors.

Table 45: Review New Course specification

2.3.6.8 <Admin> View Course List

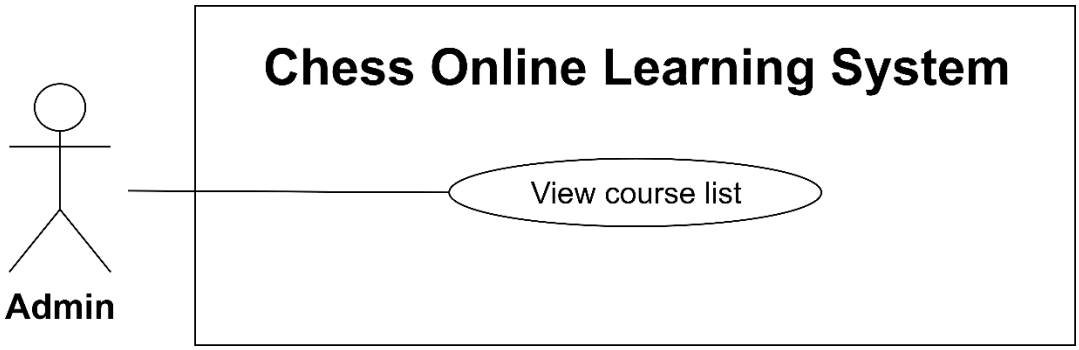


Figure 46: <Admin> View Course List

USE CASE – UC_AD.08			
Use Case No.	UC_AD.08	**Use Case Version**	2.0
Use Case Name	View Course List		
Author	Đặng Văn Hoàng		
Date	20/7/2019	**Priority**	Normal

Actor:
- Admin

Summary:
- This use case allows the admin to view the list of courses in the system.

Goal:
- Admin can view the course list and filter them by status.

Triggers:
- Admin sends the view course list command.

Preconditions:
- User signed in as an admin successfully.

Postconditions:
- Success: Course is unpublished successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin navigates to view the list of courses in the system.	System displays the course list: • Course information: name, description, the skill level required, categories, status. • Search field to search courses by name. • Select box to filter course by status. [Exception 1]

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	System does not have any course.	System course list is empty.

Relationships: N/A
Business Rules:

- Admin can view all current course in the system.
- Course list can be filtered by their status.

Table 46: View Course List specification

2.3.6.9 <Admin> Unpublish Course

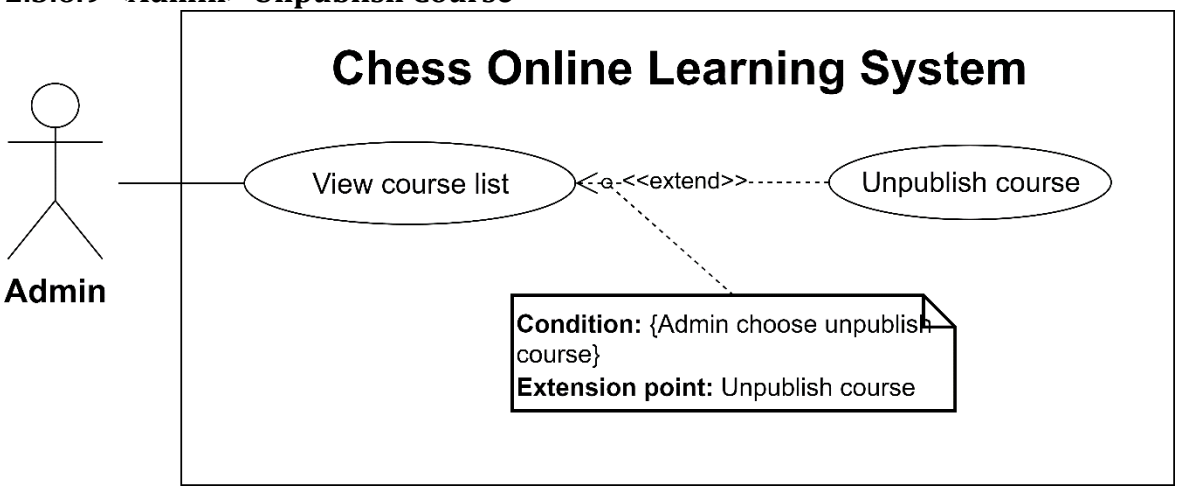


Figure 47: <Admin> Unpublish Course

USE CASE – UC_AD.09			
Use Case No.	UC_AD.09	**Use Case Version**	2.0
Use Case Name	Unpublish Course		
Author	Đặng Văn Hoàng		
Date	20/7/2019	**Priority**	Normal

Actor:
- Admin

Summary:
- This use case allows the admin to unpublish courses.

Goal:
- Admin can stop public course in the system.

Triggers:
- Admin sends the unpublish course command.

Preconditions:
- User signed in as an admin successfully.
- System must have at least 1 public course.

Postconditions:
- Success: Course is unpublished successfully.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin navigates to a specific public course.	System displays the course details: • Course information: name, description, skill level required, categories. • Course content: the number of lessons, lesson list. • Setting button
2	Admin click setting button.	System show dialog "Thay đổi trạng thái khóa học": • Ngừng công khai: button • Đóng: button
3	Admin clicks on "Ngừng công khai" button. [Alternative 1]	System shows reason field.
4	Admin inputs unpublished reason and confirm with the system.	System displays message "Thay đổi trạng thái thành công" and changes course status.

Alternative Scenario:

No	Actor Action	System Response
1	Admin clicks on "Đóng" button	System close dialogue without changing.

Exceptions: N/A
Relationships: UC_AD.08
Business Rules:

- If any course contains invalid content or bad rating, Admin can unpublish it.
- System will send a notification to instructor.
- After unpublish course, course status will transform to "Draft", course author can re-edits course and submits, then Admin will review for couse as a new course.

Table 47: Unpublish Course specification

2.3.6.10 <Admin> Review Instructor

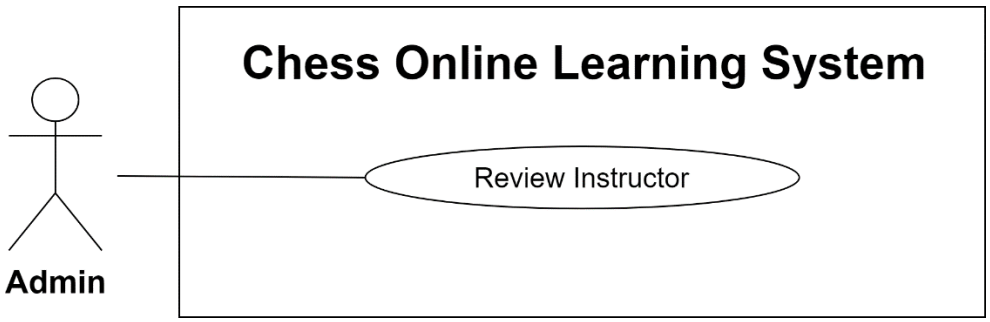


Figure 48: <Admin> Review Instructor

USE CASE – UC_AD.10			
Use Case No.	UC_AD.10	**Use Case Version**	2.0
Use Case Name	Review Instructor		

Author	Đặng Văn Hoàng		
Date	20/7/2019	Priority	Normal

Actor:
- Admin

Summary:
- This use case allows admin view the list of registered users to become instructors, view personal information of new instructor and approve or reject them.

Goal:
- Limit instructor for system.

Triggers:
- Admin sends the instructor review command.

Preconditions:
- User signed in as an Admin successfully.
- New instructor apply to system.

Post conditions:
- Success: A new instructor is approved or rejected.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response

1	Admin navigates to view new instructor.	System shows new instructor information to the form and review buttons:
		• Information fom:
		  ○ Full Name
		  ○ Email
		  ○ Achivement
		  ○ Certificate
		• Đồng ý: button
		• Từ chối: button

2	Admin clicks "Đồng ý" button [Alternative 1]	System shows a message "Người hướng dẫn mới đã được chấp nhận" and changes status of instructor.

Alternative Scenario:

No	Actor Action	System Response
1a	Admin click on "Từ chối" button	System shows rejected reason dialogue.
1b	Admin inputs rejected reason and confirm with the system.	System removes user and sends rejected email to them.

Exceptions: N/A
Relationships: N/A
Business Rules:
- Admin can review all register.
- When admin accepts, the user account can access the instructor's page. An accept notification is sent to the instructor.

• When admin rejects, the instructor is removed in the database and they can apply again.

Table 48: Review Instructor specification

2.3.6.11 <Admin> View User List

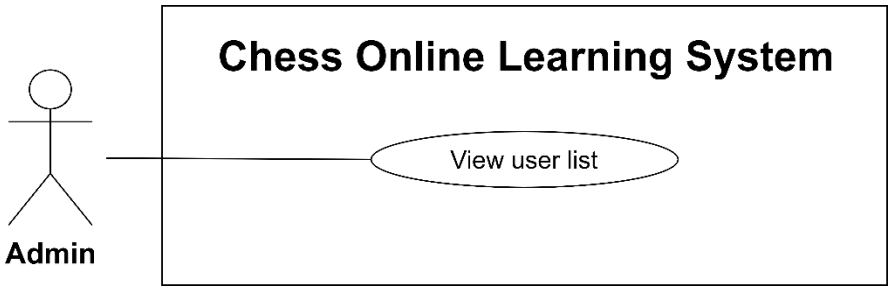


Figure 49: <Admin> View User List

USE CASE – UC_AD.11			
Use Case No.	UC_AD.11	**Use Case Version**	1.0
Use Case Name	View User List		
Author	Đặng Văn Hoàng		
Date	20/7/2019	**Priority**	Normal

Actor:
- Admin

Summary:
- This use case allows the admin view user list.

Goal:
- View list of user in the system.

Triggers:
- Admin send the view user list command.

Preconditions:
- User signed in as an admin to the system.

Postconditions:
- Success: System shows list of user in the system.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin navigates to view list of user in the system.	System shows list user information by data table with each row is: • Full Name, Email, role, Status • Change status: button (for learner and instructor).

Alternative Scenario: N/A
Exceptions: N/A
Relationships: N/A
Business Rules:
- Admin can view list of account and overview information in the system.

Table 49: View User List specification

2.3.6.12 <Admin> Active/Deactivate User

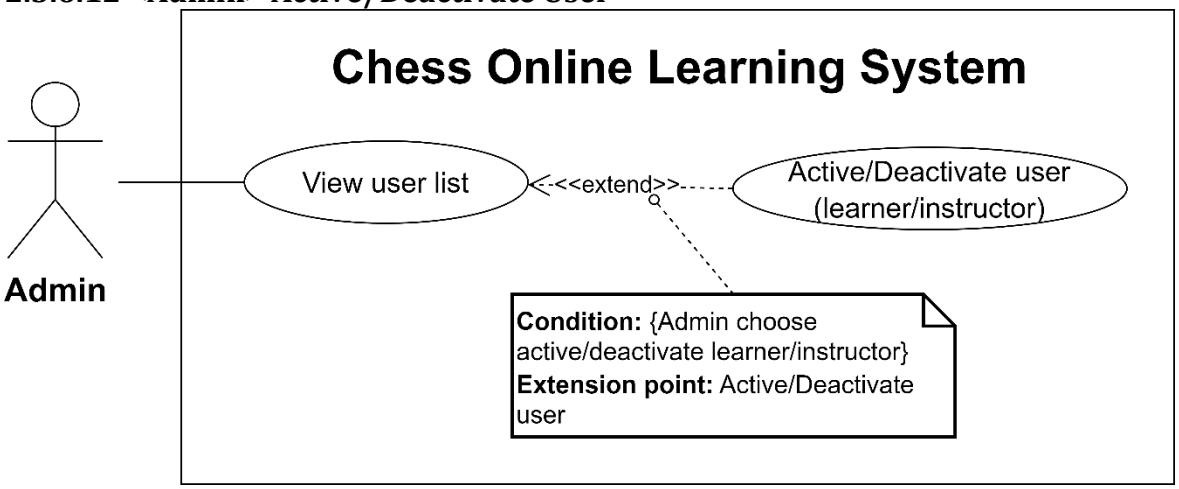


Figure 50: <Admin> Active/Deactivate User

USE CASE – UC_AD.12			
Use Case No.	UC_AD.12	**Use Case Version**	1.0
Use Case Name	Active/Deactivate User		
Author	Đặng Văn Hoàng		
Date	20/7/2019	**Priority**	Normal

Actor:
- Admin

Summary:
- This use case allows the admin to change learner or instructor status

Goal:
- Active or deactivate learner and instructor.

Triggers:
- Admin sends change user status command.

Preconditions:
- User signed in as an admin to system.

Postconditions:
- Success: System changes the status of a user.
- Failed: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin requests to active or deactivate a user.	System show dialog "Xác nhận": • Bạn có chắc chắn muốn kích hoạt (vô hiệu hóa) tài khoản <<user>>: text • Đồng ý: button • Hủy: button
2	Admin clicks on "Đồng ý" button. [Alternative 1]	System displays message "Thay đổi trạng thái thành công" and change status of user.

Alternative Scenario:

No	Actor Action	System Response

1	Admin clicks on "Hủy" button.	System closes dialogue and return to the user list.

Exceptions: N/A
Relationships: UC_AD.11
Business Rules:

- Admin can activate user account or deactivate user account.
- If admin deactivates user account, an email is sent to the user.

Table 50: Active/Deactivate User specification

3. Software System Attribute

3.1 Usability

3.1.1 Graphic User Interface

- GUI for web application is designed base on material design.
- Web application content will be written by the Vietnamese.
- Some action is represented by icon instead of text.
- Some chess fonts, icons are used to represent chess content.

3.1.2 Usability

The system usability is easy to use that users generally don't need to look at the document to use including admin and manager.

3.2 Reliability

The data should be backed up periodically.

3.3 Availability

The system must be available 24/7.

3.4 Security

- Each user have permission to access to the system by their role.
- Each signed-in user will create a session, which limits certain time in the system.

3.5 Maintainability

System is divided into independent modules.

3.6 Portability

The software itself is a web application, therefore it can be used in any platform that has a web browser and can connect to the internet.

3.7 Performance

Requests from users are responded in less than 10 seconds at 5 Mbps bandwidth speed.

4. Conceptual diagram

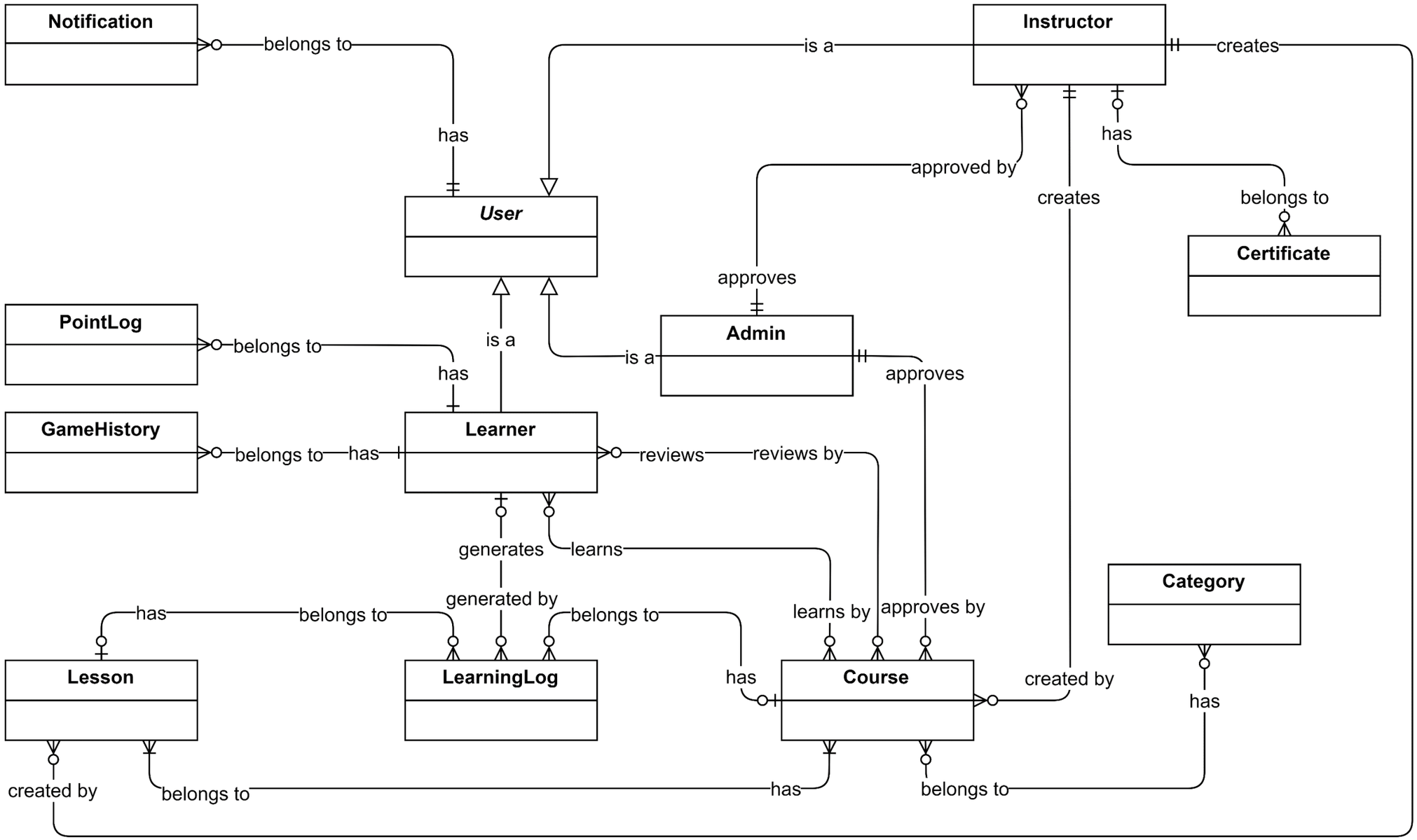


Figure 51: Conceptual diagram

ENTITY DATA DICTIONARY: DESCRIBE ALL CONTENT OF ALL ENTITIES	
Entity Name	**Description**
User	Abstract entity describes a user in the system
Notification	Contains the notification for each user
Admin	Contains the admin user information
Instructor	Contains the instructor user information
Certificate	Contains the certificates for each instructor
Learner	Contains the learner user information
GameHistory	Contain the information about the game history of learner
PointLog	Contains the point logs of learners
Course	Contains the course information
Category	Contains the system course category information
Lesson	Contains the lesson information
LearningLog	Contain the learning logs of learners

Table 51: Conceptual diagram data dictionary

D. Software Design Description

1. Design Overview

This document describes the technical and user interface design of COLS. It includes the architectural design, the detailed design of common functions and business functions and the design of the database model.

The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.

The detailed design describes the static and dynamic structure for each component and functions. It includes a class diagrams, class explanations, component diagrams, state-machines, sequence diagrams and activity diagrams for each use cases.

The database design describes the relationships between entities and the details of each entity.

Document Overview:

Section 2: gives an overall description of the system architecture design.

Section 3: gives component diagrams that describe the connection and integration of the system.

Section 4: gives the detail design description which includes class diagram, class explanation, state-machines, sequence diagrams and activity diagrams to details the application functions.

Section 5: describe screens design.

Section 6: describe a fully attributed ERD.

Section 7: describe algorithms.

2. System Architectural Design

2.1 System Architecture Overview[2]

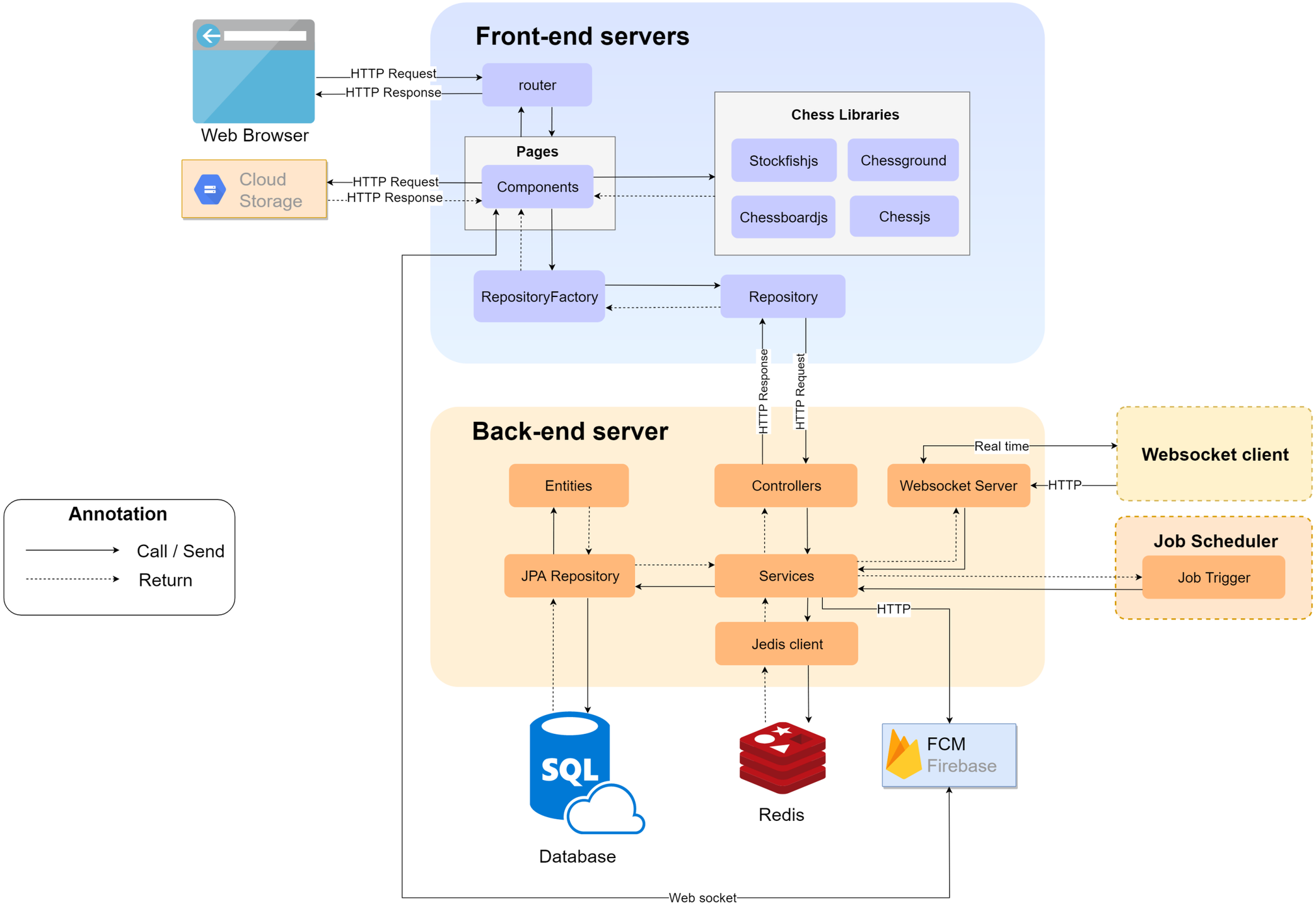


Figure 52: System architecture overview

This system is developed base on multi-tiers architecture with Database, Back-end server for process business logic and 2 front-end servers (for learning-app and admin-app) to process presentation.

a. Data tier

In Data tier, we use Relational Database Management System - MySQL to store system data.

b. Business Logic tier

In Business Logic tier, we use Spring Boot to develop RESTful APIs with MVC pattern.

In this system, the chess engine performs in the presentation tier, then we using websocket to synchronize game states between server and client.

We also use Redis, an in-memory data structure for caching data.

In order to schedule automatic tasks such as giving course suggestions to learners. We use Quartz Job 2.0 to schedule the task to run repeatedly. Besides, it is also used to trigger the timeout of the chess game by the cron expression.

c. Presentation tier

For Front-end applications, we use VueJS framework and Repository-pattern to manage data. We also apply chess libraries to solve our problems, which be related to user interaction, include:

- **Stockfishjs:** Stockfish is an opensource chess engine, which supports to analyse chess move and Stockfishjs is a pure JavaScript implementation of Stockfish. We can use UCI (Universal Chess Interface) to communicate with the chess engine. The chess engine performs as a worker thread, they receive UCI command by `postMessage()` method and perform tasks to get a result, and return the result by `onmessage()` method. We apply stockfishjs to build chess bot, which can autoplay with learner and support learner in exercises.
- **Chessjs:** Chessjs is a chess library which use to generate chess move and validate the client move. We use this library to validate user move and manage current status of chess game, such as turn colour, pieces position.
- **Chessboardjs and Chessground:** There 2 library also support to render Graphic User Interface (GUI), 2 libraries which be discussed above only support data in console, then we must use this libraries to display chessboard for the user. They can perform move and display pieces position from Chessjs and Stockfishjs by using FEN String, which will be discussed below.

d. Cloud Services

In this system, we use cloud services, which are Cloud Storage and Firebase Cloud Messaging (FCM). We use Cloud Storage of Google Cloud Platform to store images which are uploaded by the client and store the URL in Database. The FCM is used to push real-time notifications easily and conveniently.

3. Component Diagram[1]

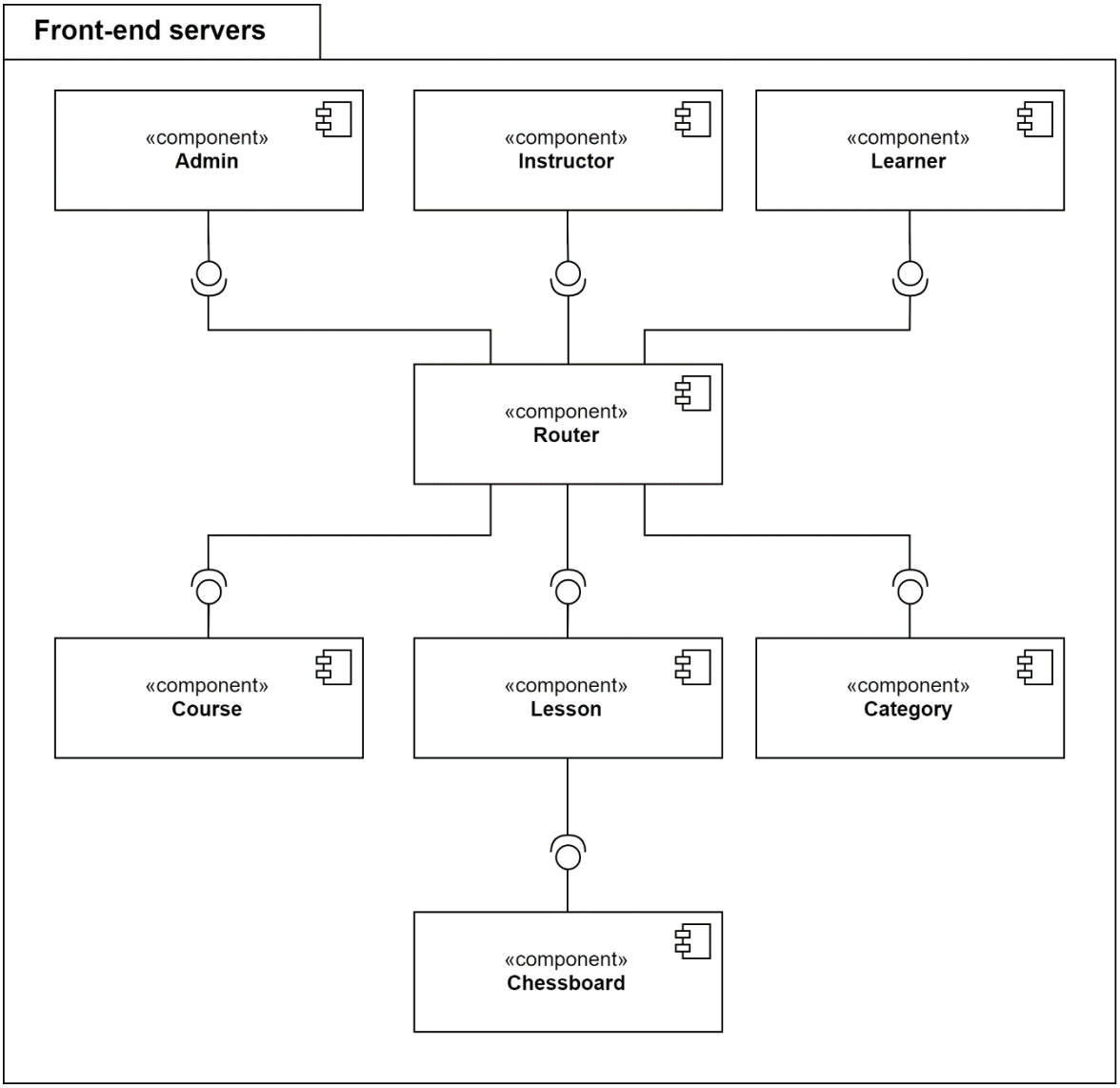


Figure 53: Front-end Servers Component diagram

COMPONENT DICTIONARY: DESCRIBES COMPONENTS	
Router	Handle HTTP request, response
Course Component	Handle activity about Course
Lesson Component	Handle activity about Lesson
Chessboard Component	Handle activity about Chessboard
Category Component	Handle activity about Category
Admin Component	Handle activity of Admin
Instructor Component	Handle activity of Instructor
Learner Component	Handle activity of Learner

Table 52: Front-end Servers Component Dictionary

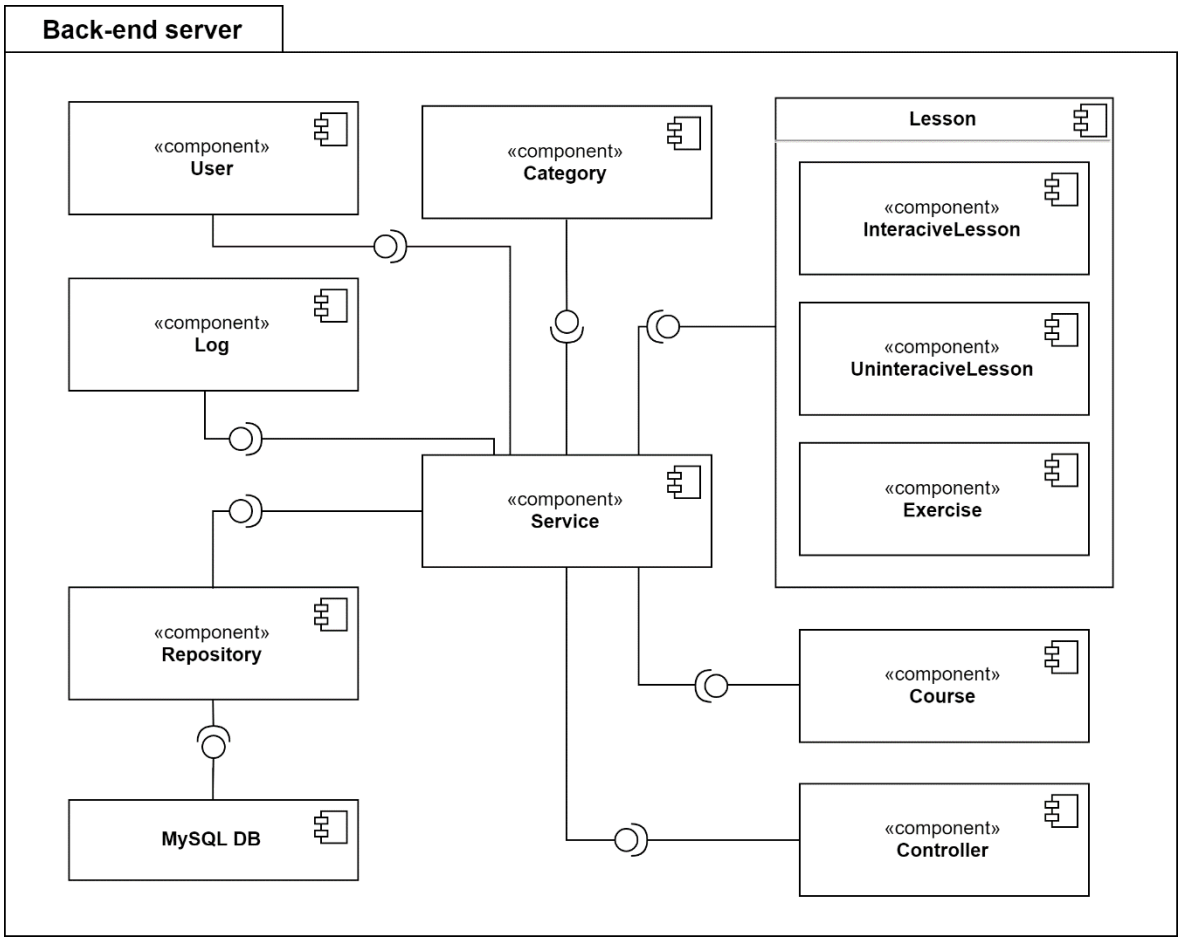


Figure 54: Back-end Server Component diagram

COMPONENT DICTIONARY: DESCRIBES COMPONENTS	
Constroller	Handle HTTP request, response
Service Component	Process data
Repository Component	Access data from database
MySQL DB	Database component
Category Component	Handle activity about Category
User Component	Handle activity of User
Log Component	Handle log about system activity
Course Component	Handle activity about Course
Lesson Component	Handle activity about Lesson
Interactive Lesson Component	Handle activity about Lesson (type Interactive)
Uninteractive Lesson Component	Handle activity about Lesson (type Uninteractive)
Exercise Component	Handle activity about Exercise

Table 53: Back-end Servers Component Dictionary

4. Detail Description

4.1 Class Diagram[1]

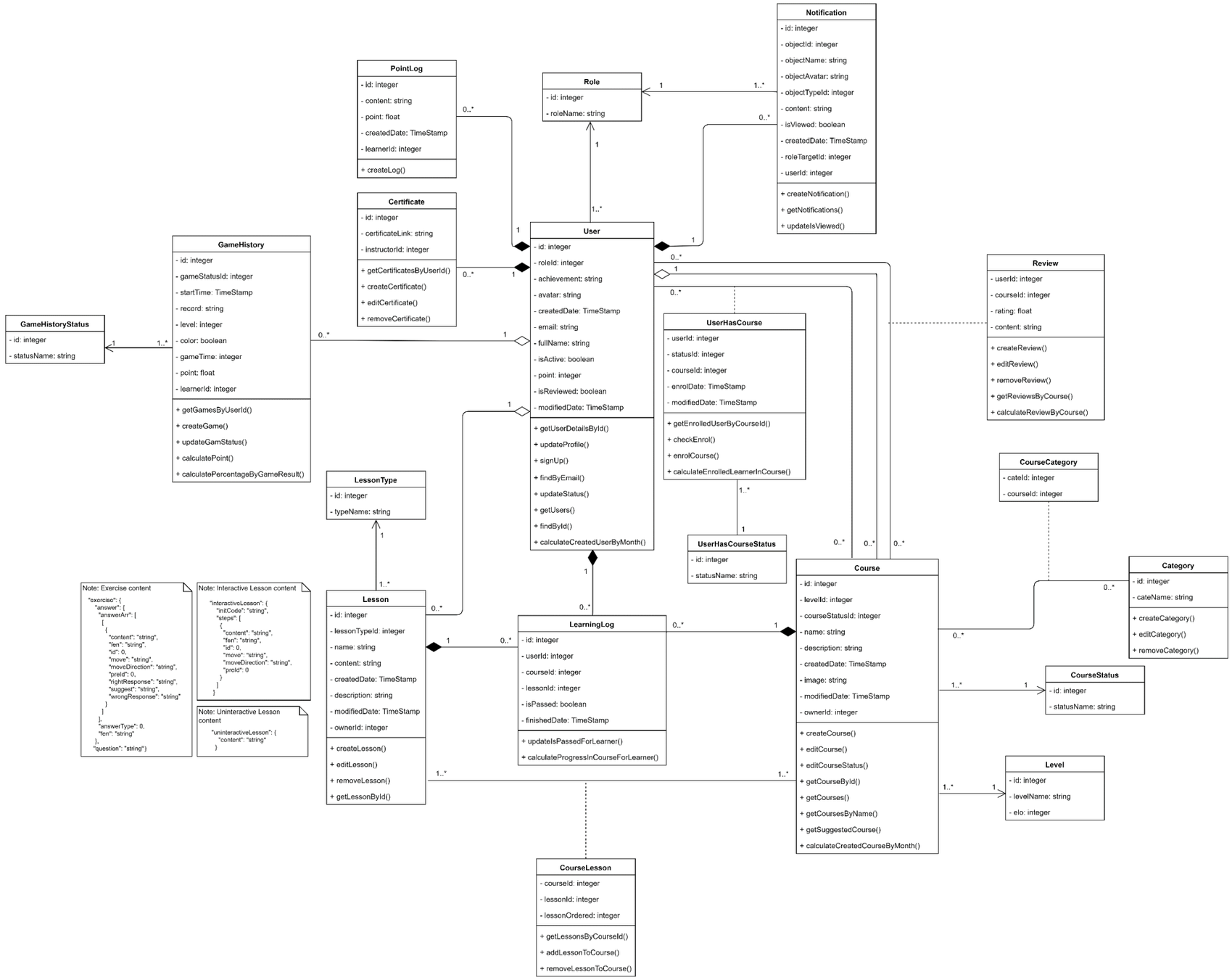


Figure 55: Class Diagram

CLASS DICTIONARY: DESCRIBE CLASS		
Class Name	**Mapping column with Conceptual diagram**	**Description**
User	User	Contains the user information
Role	N/A	Not exist in Conceptual diagram but need this class to contain role name for user
Notification	Notification	Contains the notification for user
Certificate	Certificate	Contains the certificate information for instructor
PointLog	N/A	Not exist in Conceptual diagram but need this class to log point transaction for learner
GameHistory	GameHistory	Contain the game history information for learner
GameHistoryStatus	N/A	Not exist in Conceptual diagram but need this class to contain the status name of game history
Course	Course	Contains the course information
CourseStatus	N/A	Not exist in Conceptual diagram but need this class to contain status name of course
CourseLesson	N/A	Not exist in Conceptual diagram but need this class to contain the course which lesson belongs to
Review	Review	Contains the review for each course by learner
UserHasCourse	N/A	Not exist in Conceptual diagram but need this class to contain status for each course of learner
UserHasCourseStatus	N/A	Not exist in Conceptual diagram but need this class to contain status name for each course of learner
Category	Category	Contains the category information
CourseCategory	N/A	Not exist in Conceptual diagram but need this class to contain the relationship between courses and categories
Lesson	Lesson	Contains the lesson information

LessonType	N/A	Not exist in Conceptual diagram but need this class to contain the type name for each lesson
LearningLog	LearningLog	Contain learning logs of learners

Table 54: Class Dictionary

4.2 Class Diagram Explanation

4.2.1 User

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a user
roleId	Integer	Private	Role identifier to map with role name
achievement	String	Private	Instructor achievement
avatar	String	Private	User's avatar URL
createdDate	TimeStamp	Private	Account's created time
email	String	Private	User's email
fullName	String	Private	User's fullName
isActive	Boolean	Private	User's status
point	Float	Private	Individual point of learner
isReviewed	Boolean	Private	Instructor's review status
modifiedDate	TimeStamp	Private	User's information updated time
Method	**Return Type**	**Visibility**	**Description**
getUserDetailsById	Object	Public	Get details of the user by identifier
updateProfile	Object	Public	Update profile for the user
signUp	Boolean	Public	Sign up new account
findByEmail	Boolean	Public	Find a user by email
updateStatus	Object	Public	Update status of the user
getUsers	List	Public	Get list of user
findById	Boolean	Public	Find a user by Identifier
calculateCreated UserByMonth	Integer	Pulic	Calculate number of new user by month

Table 55:User attributes

4.2.2 Role

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a role
roleName	String	Private	Role's name

Table 56: Role attributes

4.2.3 Notification

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a Notification
objectId	Integer	Private	Identifier of notification object

objectName	String	Private	Notification object's name
objectAvatar	String	Private	Notification object's image
objectTypeId	Integer	Private	Notification object's type identifier
content	String	Private	Notification content
isViewed	Boolean	Private	Tracking notification is view or not
createdDate	TimeStamp	Private	Notification created time
roleTargetId	Integer	Private	Target user role identifier
userId	Integer	Private	Target user identifier
Method	**Return Type**	**Visibility**	**Description**
createNotification	Object	Public	Create a notification
getNotifications	List	Public	Get list of notification
updateIsViewed	Object	Public	Update the notification to viewed

Table 57: Notification attributes

4.2.4 Certificate

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a Certificate
certificateLink	String	Private	Certificate image's URL
instructorId	Integer	Private	Owner's identifier
Method	**Return Type**	**Visibility**	**Description**
getCertificatesBy UserId	List	Public	Get list of certification of a user by user identifier
createCertificate	Object	Public	Create a certificate
editCertificate	Object	Public	Edit a certificate
removeCertificate	Boolean	Public	Remove a certificate

Table 58: Certificate attributes

4.2.5 PointLog

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a log
content	String	Private	Description for point transaction
point	Float	Private	Number of point
createdDate	Timestamp	Private	Transaction's time
learnerId	Integer	Praivet	Transaction owner's identifier
Method	**Return Type**	**Visibility**	**Description**
createLog	Object	Public	Create a point log

Table 59: PointLog attributes

4.2.6 GameHistory

Attribute	Type	Visibility	Description
id	Long	Private	Unique identifier of a Game History

gameStatusId	Integer	Private	Game state's identifier to map with state name
startTime	TimeStamp	Private	The time when game be started
record	String	Private	Game content
level	Integer	Private	Bot's level which learner choose
color	Boolean	Private	Learner colour
gameTime	Integer	Private	Game Time duration
point	Float	Private	Bonus point for learner when ending game
learnerId	Integer	Private	Game owner's identifier
Method	**Return Type**	**Visibility**	**Description**
getGamesByUserId	List	Public	Get list of games information by user identifier
createGame	Object	Public	Create a chess game
updateGameStatus	Object	Public	Update game status
calculatePoint	Integer	Public	Calculate bonus point when ending game
calculatePercentage ByGameResult	Array	Public	Calculate percentage for each game result

Table 60: GameHistory attributes

4.2.7 GameHistoryStatus

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a Game history status
statusName	String	Private	Game history status's name

Table 61: GameHistoryStatus attributes

4.2.8 Course

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a Course
levelId	Integer	Private	Identifier of course level to map with level name
courseStatusId	Integer	Private	Identifier of course status to map with status name
name	String	Private	Course's name
description	String	Private	Course's description
createdDate	Timestamp	Private	Course's created time
image	String	Private	Course's cover image URL
modifiedDate	TimeStamp	Private	Course's information updated time
ownerId	Integer	Private	Course owner's identifier
Method	**Return Type**	**Visibility**	**Description**
createCourse	Object	Public	Create a course

editCourse	Object	Public	Update course information
editCourseStatus	Boolean	Public	Update course status
getCourseById	Object	Public	Get a course by identifier
getCourses	List	Public	Get list of course
getCoursesBy Name	List	Public	Get courses which contain name keyword
getSuggested Course	List	Public	Get list of suggested course
calculateCreated CourseByMonth	List	Public	Calculate number of new course for each month

Table 62: Course attributes

4.2.9 CourseStatus

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a course status
statusName	String	Private	Course's status name

Table 63: CourseStatus attributes

4.2.10 CourseLesson

Attribute	Type	Visibility	Description
courseId	Integer	Private	Identifier of couse which has lesson
lessonId	Integer	Private	Identifier of lesson which belongs to course
lessonOrdered	Integer	Private	Ordered number of lesson in course
Method	**Return Type**	**Visibility**	**Description**
getLessonsBy CourseId	List	Public	Get list of lesson in a course by course identifier
addLesson ToCourse	Object	Public	Add a lesson to a course
removeLesson ToCourse	Object	Public	Remove a lesson out of a course

Table 64: CourseLesson attributes

4.2.11 Review

Attribute	Type	Visibility	Description
userId	Integer	Private	Identifer of learner who created review
courseId	Integer	Private	Identifier of course that review belongs to
rating	Float	Private	Rating point
content	String	Private	Rating content
Method	**Return Type**	**Visibility**	**Description**
createReview	Object	Public	Create a review

editReview	Object	Public	Edit a review
removeReview	Boolean	Public	Remove a review
getReviewsBy Course	List	Public	Get list of review in course by course identifier
calculateReview ByCourse	Float	Public	Calculate average review for course

Table 65: Review attributes

4.2.12 UserHasCourse

Attribute	Type	Visibility	Description
userId	Integer	Private	Identifier of the user who has a relationship with the course
statusId	Integer	Private	Identifier of status, which is the relationship between course and user and map with a status name.
coursed	Integer	Private	Identifier of course which belongs to the user
enrolDate	TimeStamp	Private	Enrol Time of learner with course
modifiedDate	TimeStamp	Private	Relationship's status updated time
Method	**Return Type**	**Visibility**	**Description**
getEnrolledUser ByCourseId	Integer	Public	Get number of enrolled learner in a course by course identifier
checkEnrol	Boolean	Public	Check course is enrolled by learner
enrolCourse	Boolean	Public	Learner enrol a course
calculateEnrolled LearnerInCourse	Integer	Public	Calculate number of enrolled learner in course

Table 66: UserHasCourse attributes

4.2.13 UserHasCourseStatus

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of status
statusName	String	Private	Status's name

Table 67: UserHasCourseStatus attributes

4.2.14 Category

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a category
cateName	String	Private	Category's Name
Method	**Return Type**	**Visibility**	**Description**
createCategory	Object	Public	Create a category
editCategory	Object	Public	Edit a category
removeCategory	Object	Public	Remove a category

Table 68: Category discussion

4.2.15 CourseCategory

Attribute	Type	Visibility	Description
cateId	Integer	Private	Identifier of category which has course
courseId	Integer	Private	Identifier of course which belongs to category

Table 69: CourseCategory attributes

4.2.16 Lesson

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a lesson
lessonTypeId	Integer	Privte	Unique identifier of lesson type to map with type name
name	String	Private	Lesson's name
content	String	Private	Lesson's content
createdDate	TimeStamp	Private	Lesson's created time
description	String	Private	Lesson's description
modifiedDate	TimeStamp	Private	Lesson's information updated time
ownerId	Integer	Private	Lesson owner's identifier
Method	**Return Type**	**Visibility**	**Description**
createLesson	Object	Public	Create a lesson
editLesson	Object	Public	Edit a lesson
removeLesson	Boolean	Public	Remove a lesson
getLessonById	Object	Public	Get a lesson by identifier

Table 70: Lesson attributes

4.2.17 LessonType

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a type of lesson
typeName	String	Private	Name of lesson type

Table 71: LessonType attributes

4.2.18 LearningLog

Attribute	Type	Visibility	Description
id	Integer	Private	Unique identifier of a learning log
userId	Integer	Private	Identifier of learner who learn the lesson
coursed	Integer	Private	Identifier of course which lesson belongs to
lessonId	Integer	Private	Identifier of lesson which learner learn
isPassed	Boolean	Private	Tracking lesson is finished or not
finishedDate	TimeStamp	Private	Lesson's finished time

Method	Return type	Visibility	Description
updateIsPassed ForLearner	Boolean	Public	Update lesson is passed by the learner
calculateProgress InCourseFor Learner	Integer	Public	Calcualte learning progress in course for learner

Table 72: LearningLog attributes

4.3 State Machines[1]

4.3.1 Course State

Summary: this diagram shows state machines of a course in the system

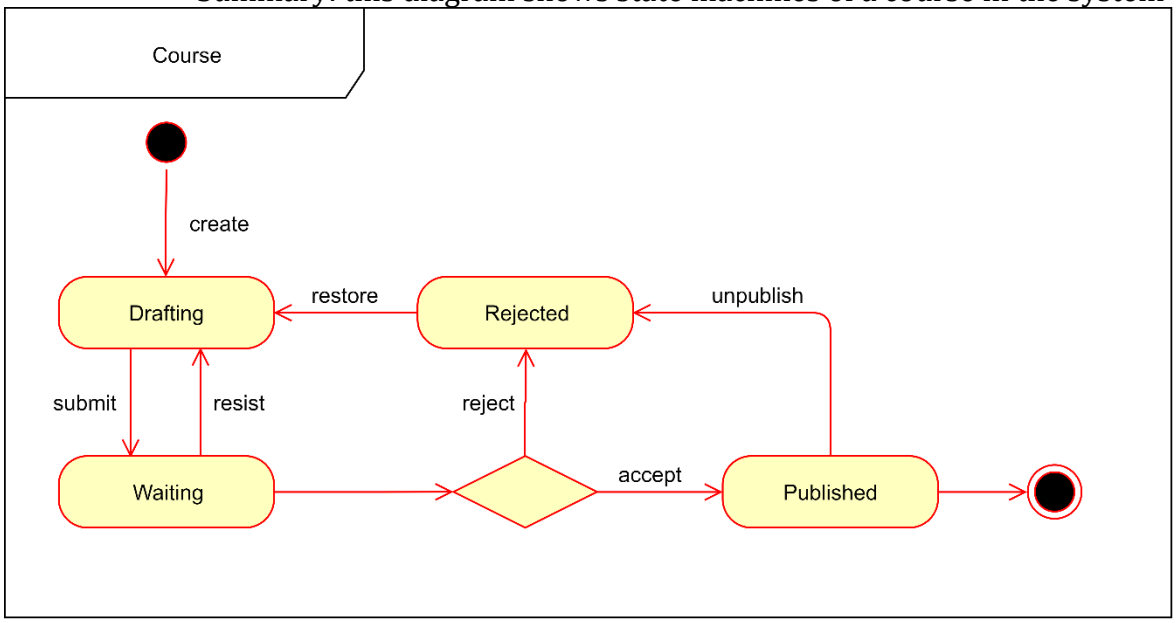


Figure 56: Course State

4.3.2 Learner Has Course State

Summary: this diagram shows state machines of a relationship between learner and course in system

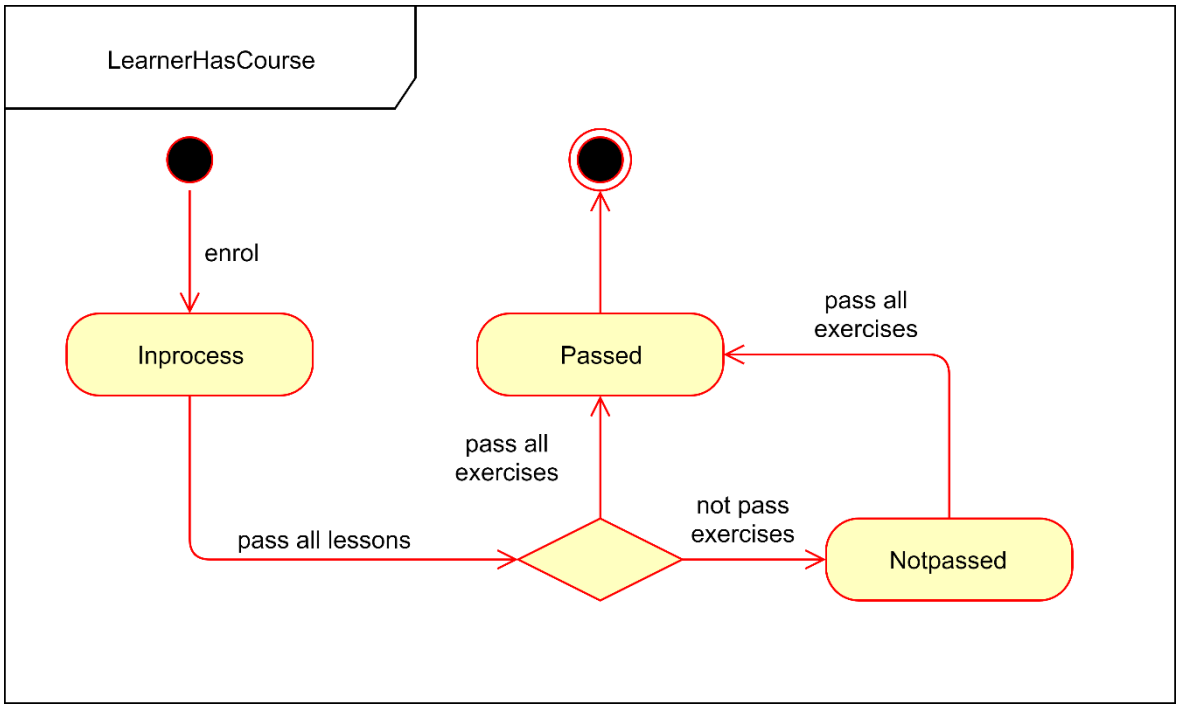


Figure 57: Learner Has Course State

4.4 Interactive Diagram[1]

4.4.1 Guest & Learner

4.4.1.1 View Published Course

Summary: this diagram shows process of view published course.

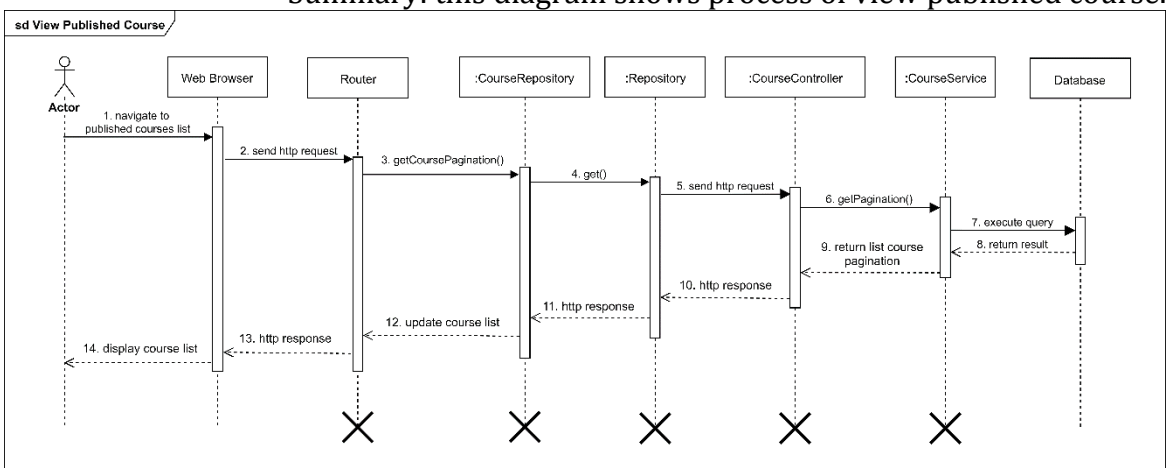


Figure 58: Sequence Diagram - <Guest & Learner> View Published Course

4.4.1.2 Search Courses

Summary: this diagram shows process of search courses by name.

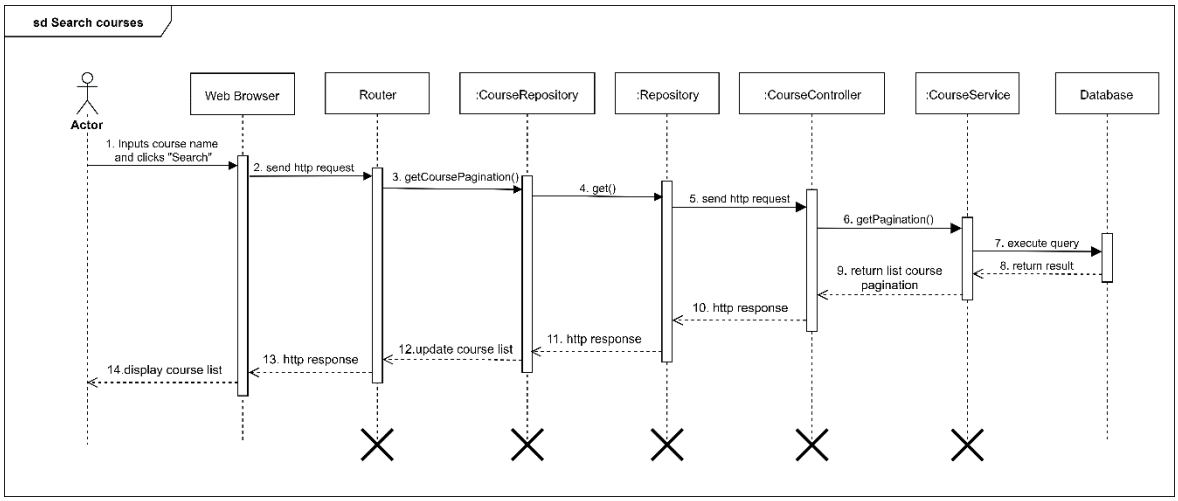


Figure 59: Sequence Diagram - <Guest & Learner> Search Courses

4.4.1.2 View Course Details

Summary: this diagram shows process of view course details.

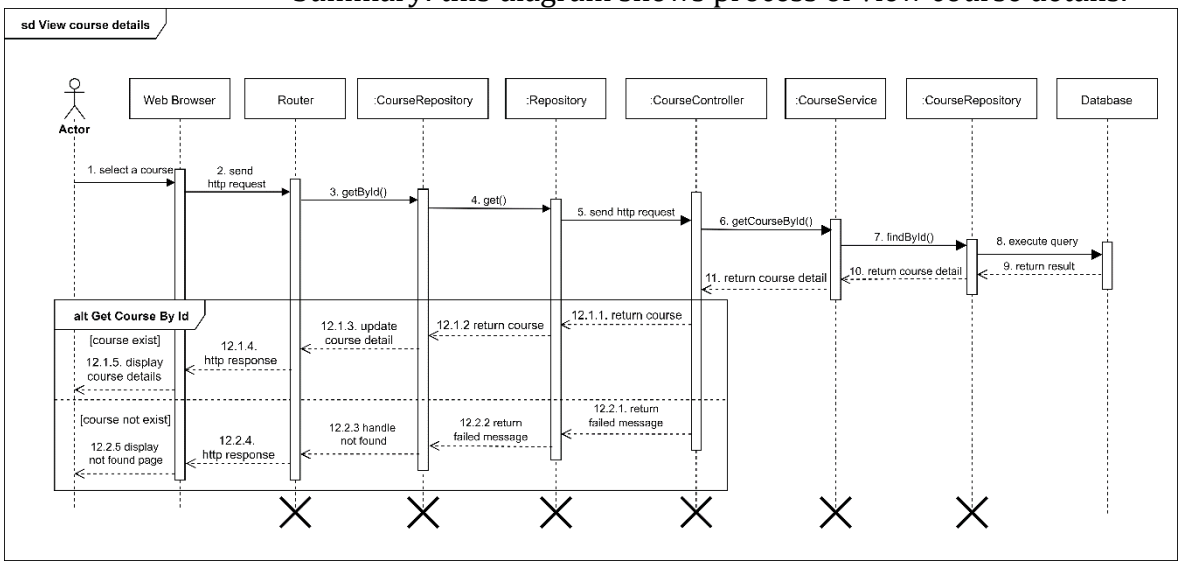


Figure 60: Sequence Diagram - <Guest & Learner> View Course Details

4.4.2 Learner

4.4.2.1 Enrol Course

Summary: this diagram shows the process to enrol a course

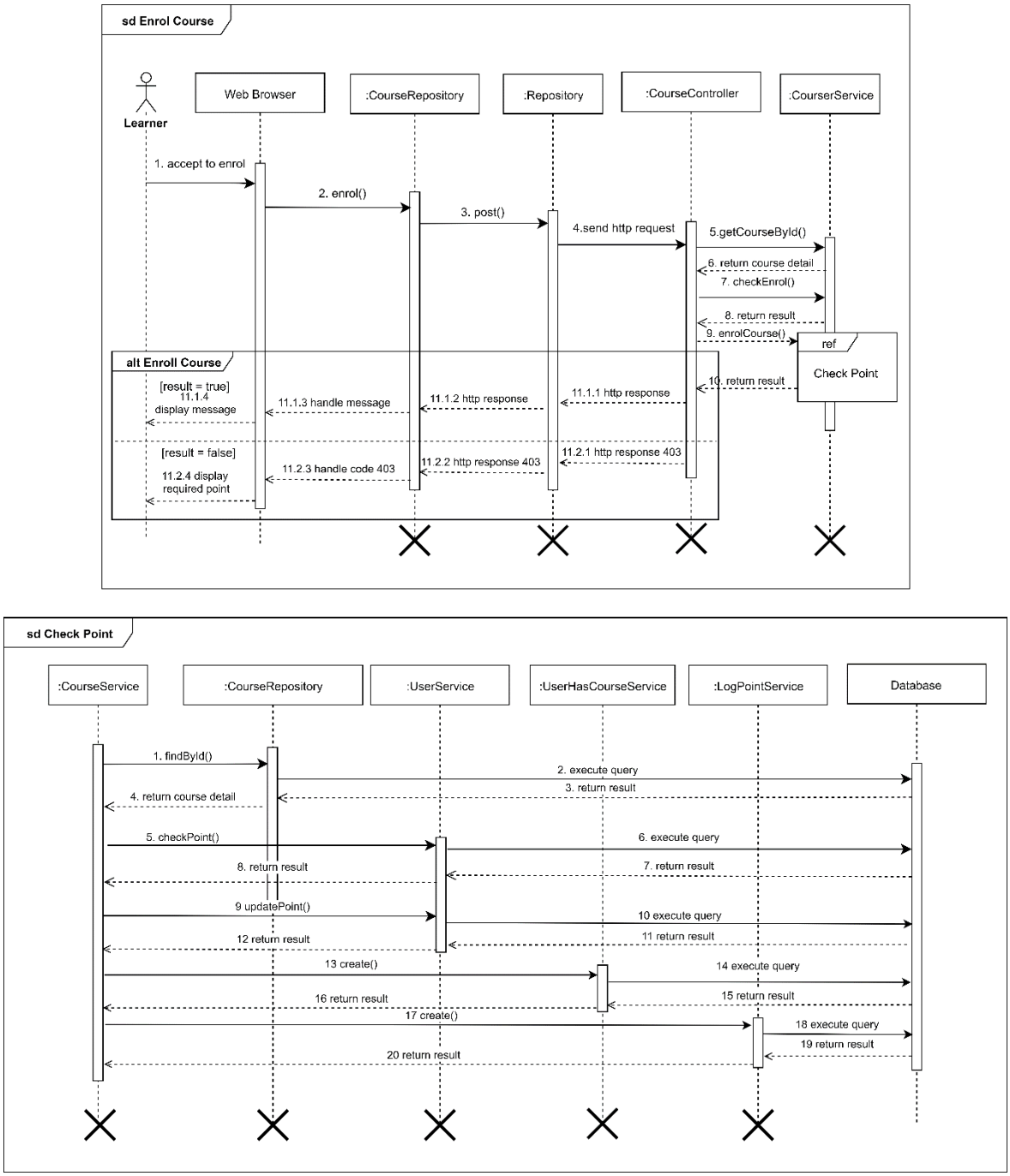


Figure 61: Sequence Diagram - <Learner> Enrol Course

4.4.2.2 Create Review

Summary: this diagram shows process of create a review

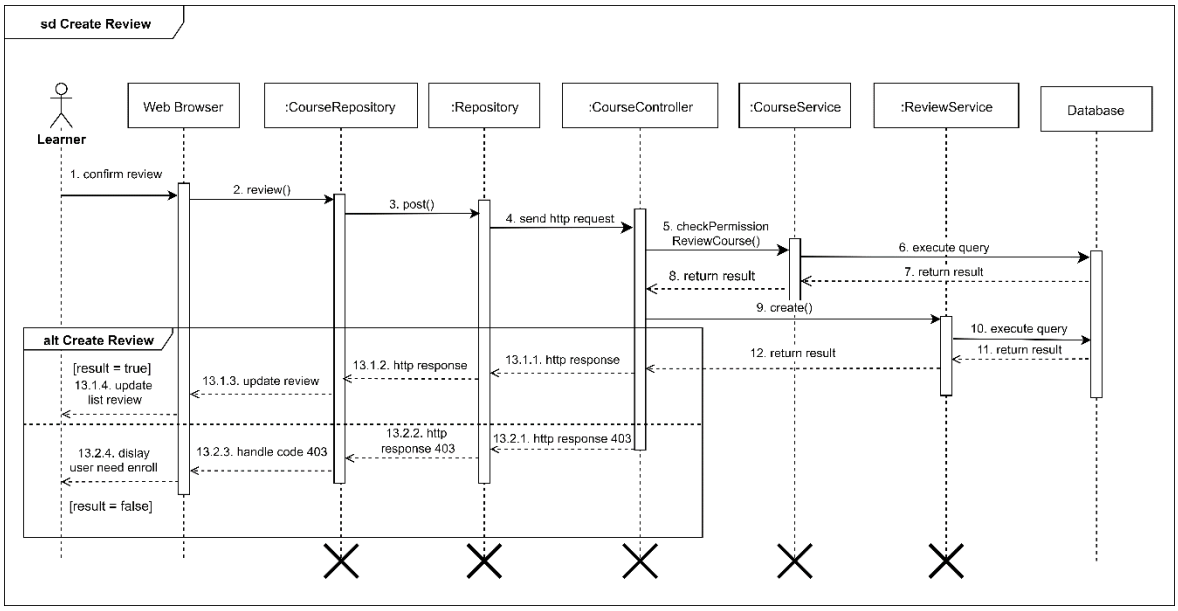


Figure 62: Sequence Diagram - <Learner> Create Review

4.4.2.3 View List Of Suggested Course

Summary: this diagram shows process of view list of suggested course

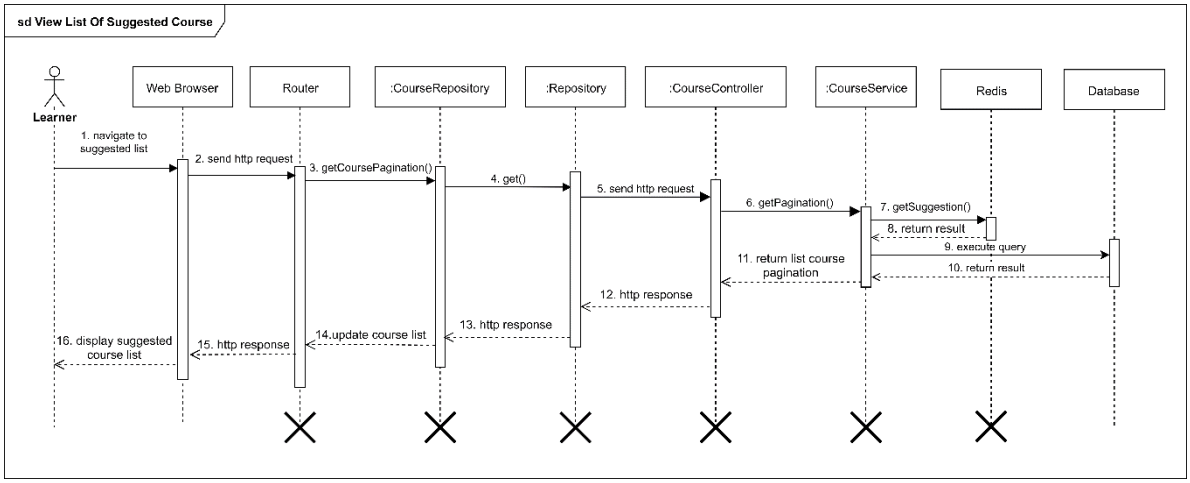
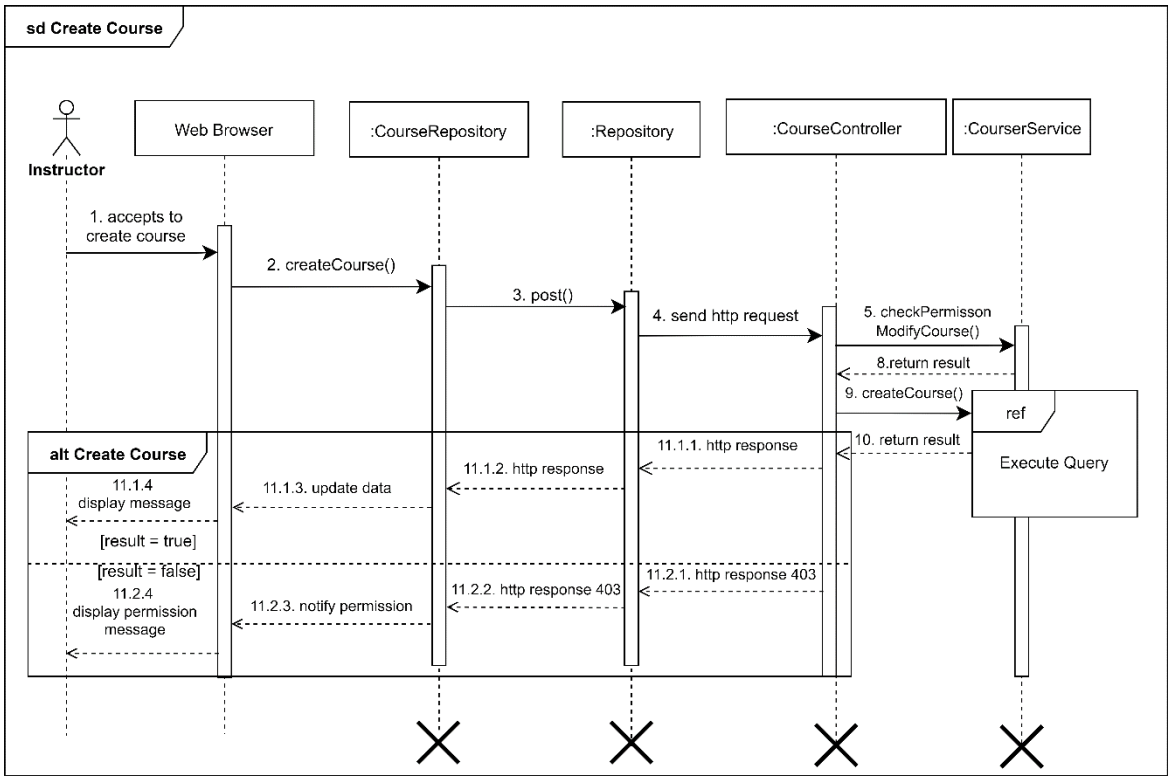


Figure 63: Sequence Diagram - <Learner> View List Of Suggested Course

4.4.3 Instructor

4.4.3.1 Create Course

Summary: this diagram shows process of create a course



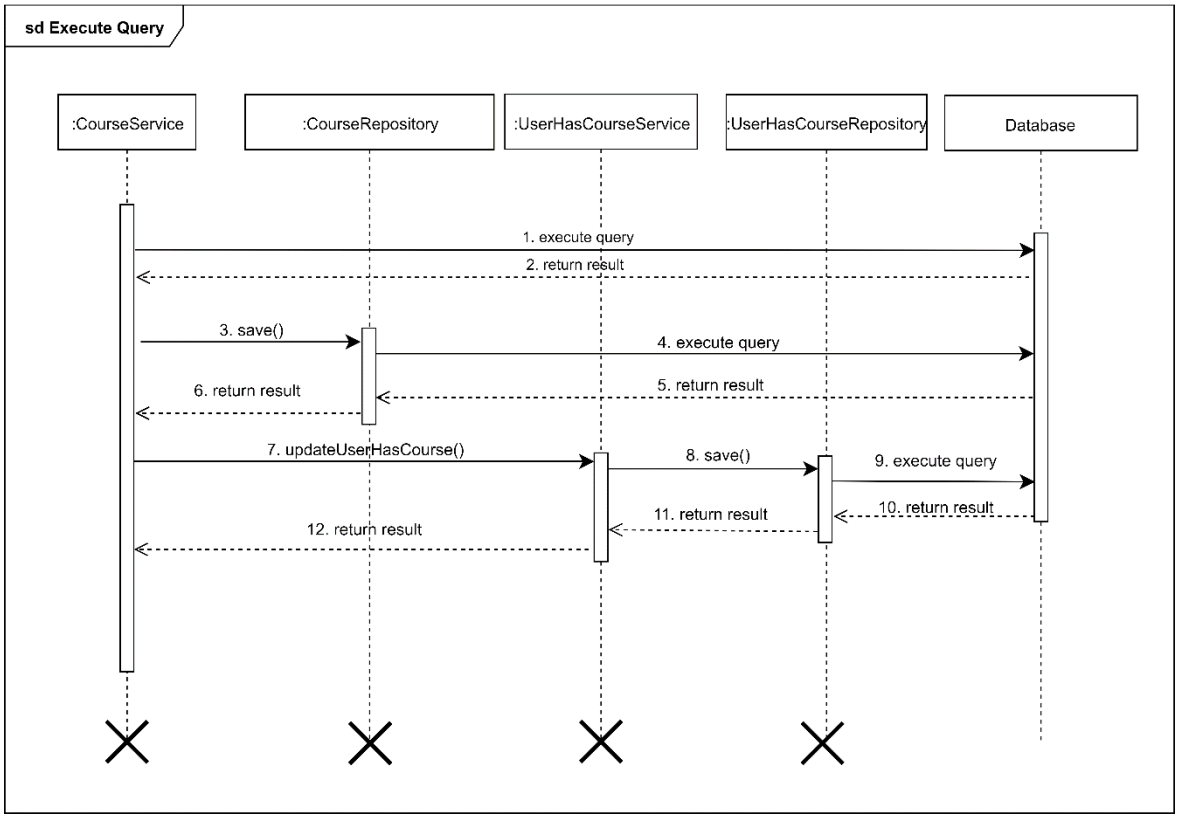
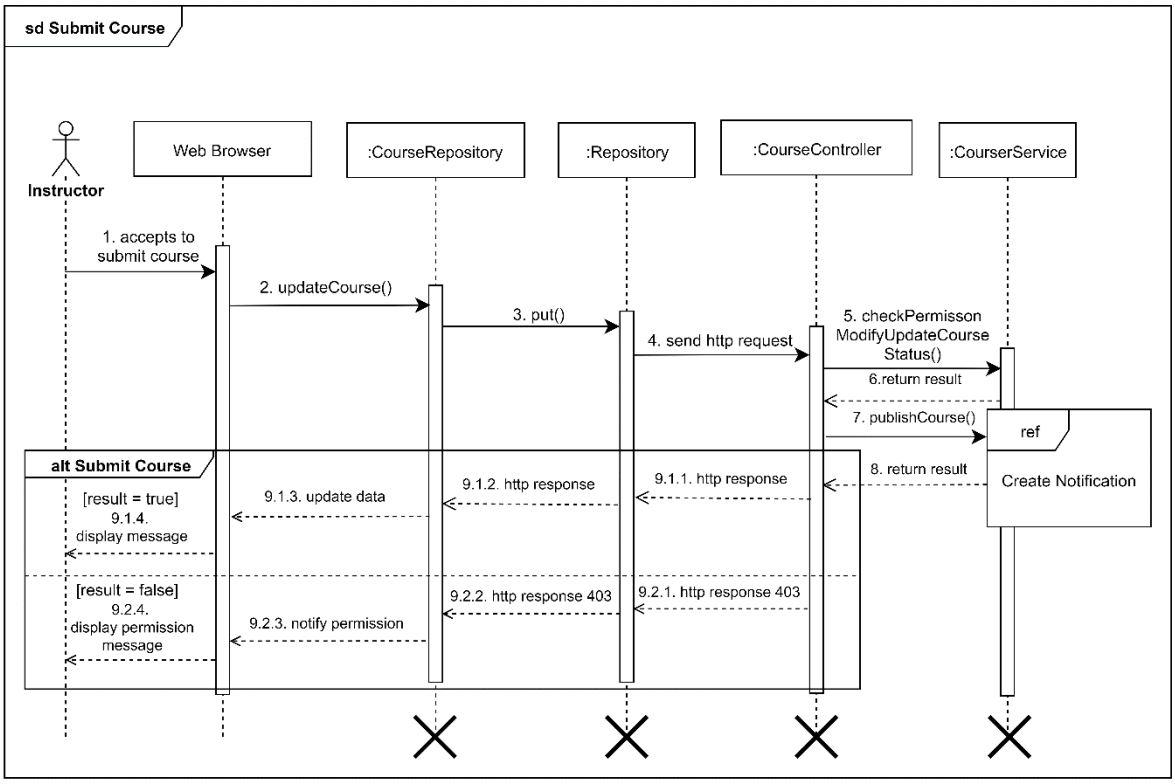


Figure 64: Sequence Diagram - <Instructor> Create Course

4.4.3.2 Submit Course

Summary: this diagram shows process of submit course



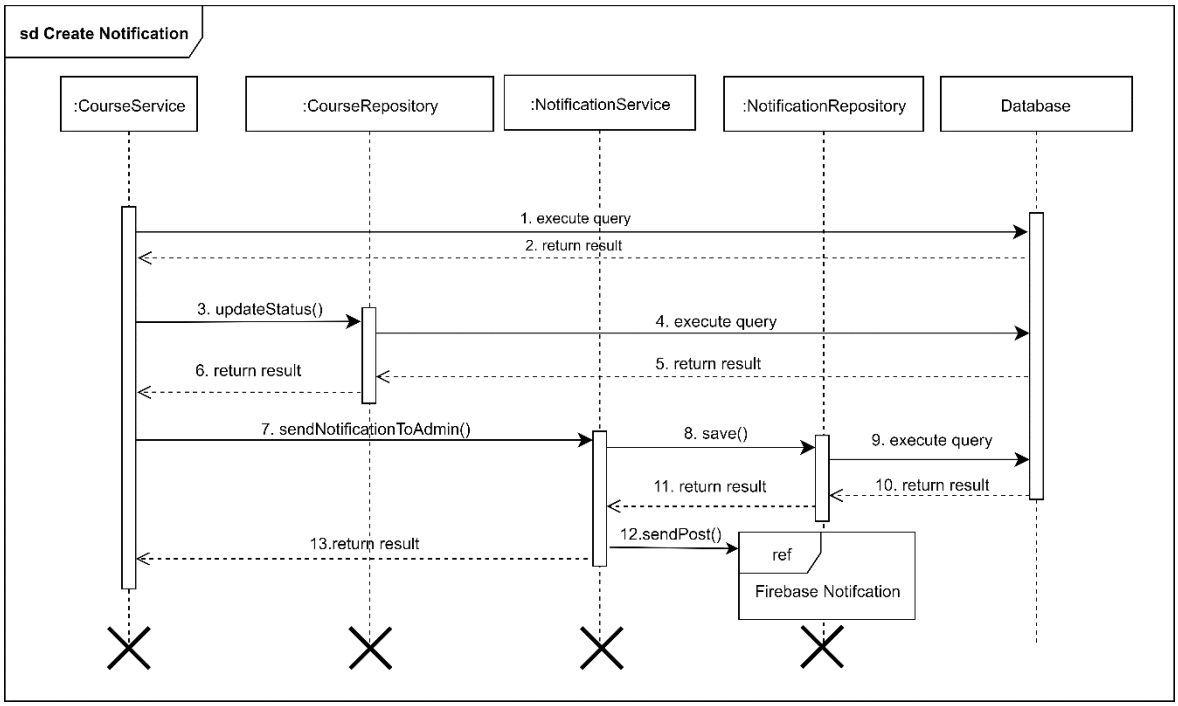
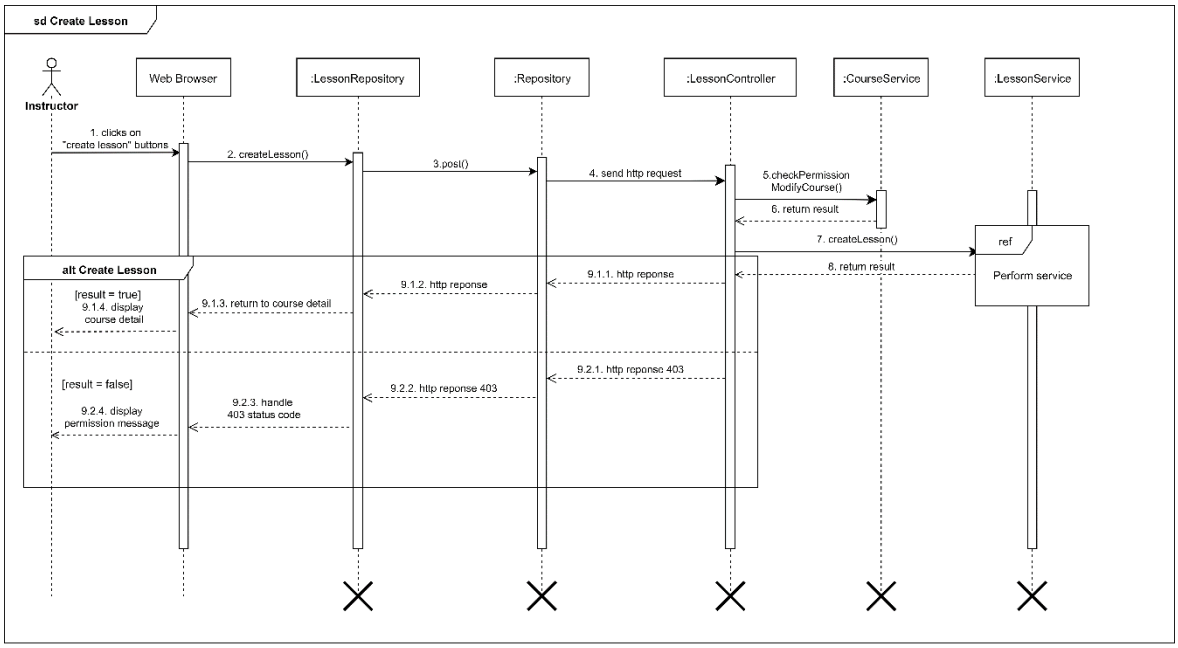


Figure 65: Sequence Diagram - <Instructor> Submit Course

4.4.3.3 Create Lesson

Summary: this diagram shows process of creating lesson or exersise



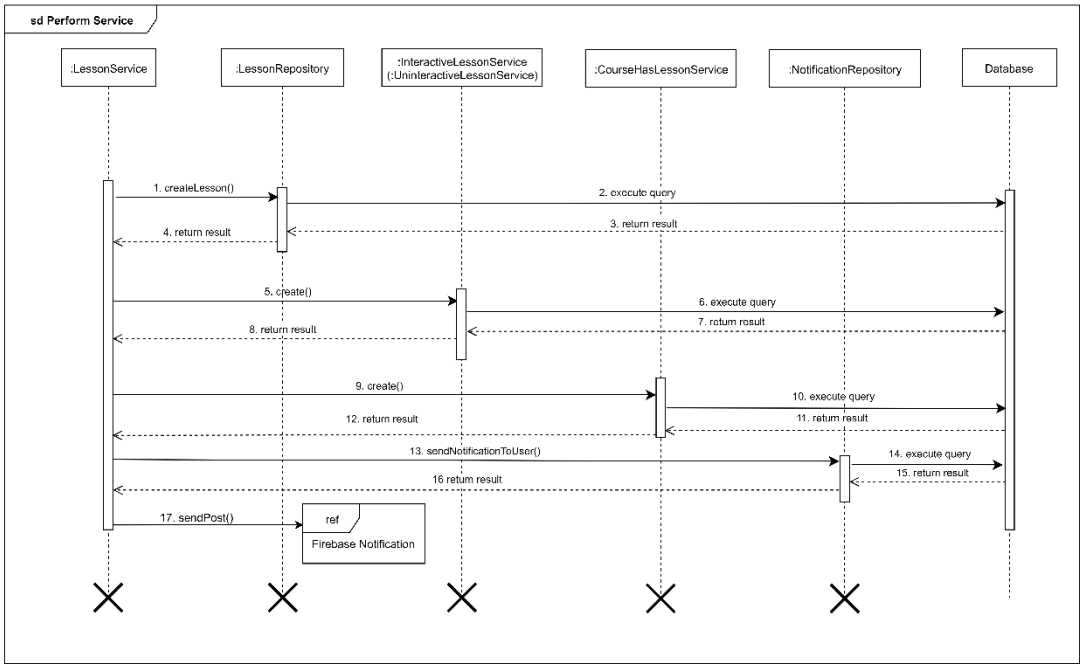
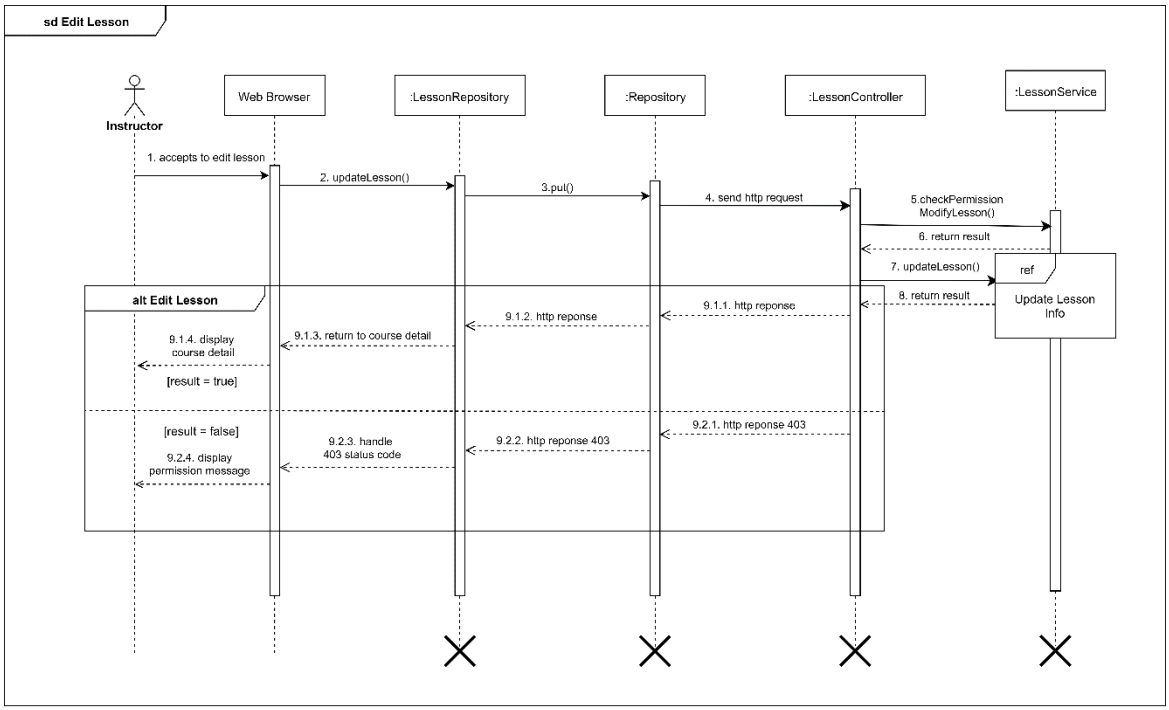


Figure 66: Sequence Diagram - <Instructor> Create Lesson

4.4.3.4 Edit Lesson

Summary: this diagram shows process of editing lesson or exersise



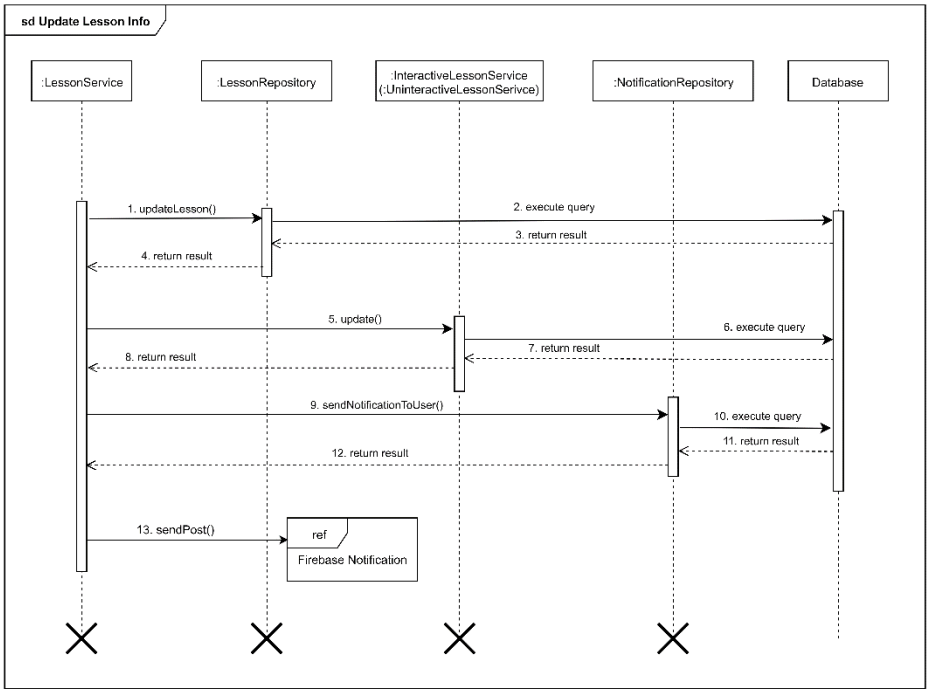
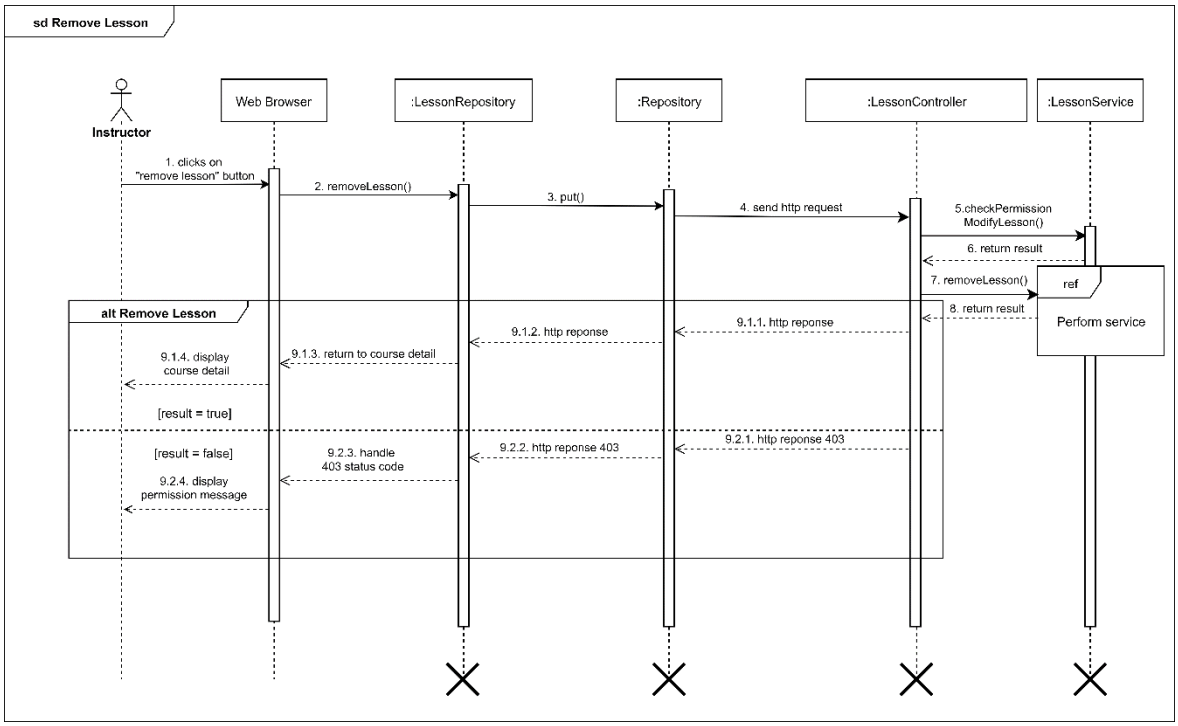


Figure 67: Sequence Diagram - <Instructor> Edit Lesson

4.4.4.6 Remove Lesson

Summary: this diagram shows process of removing a lesson or exercise



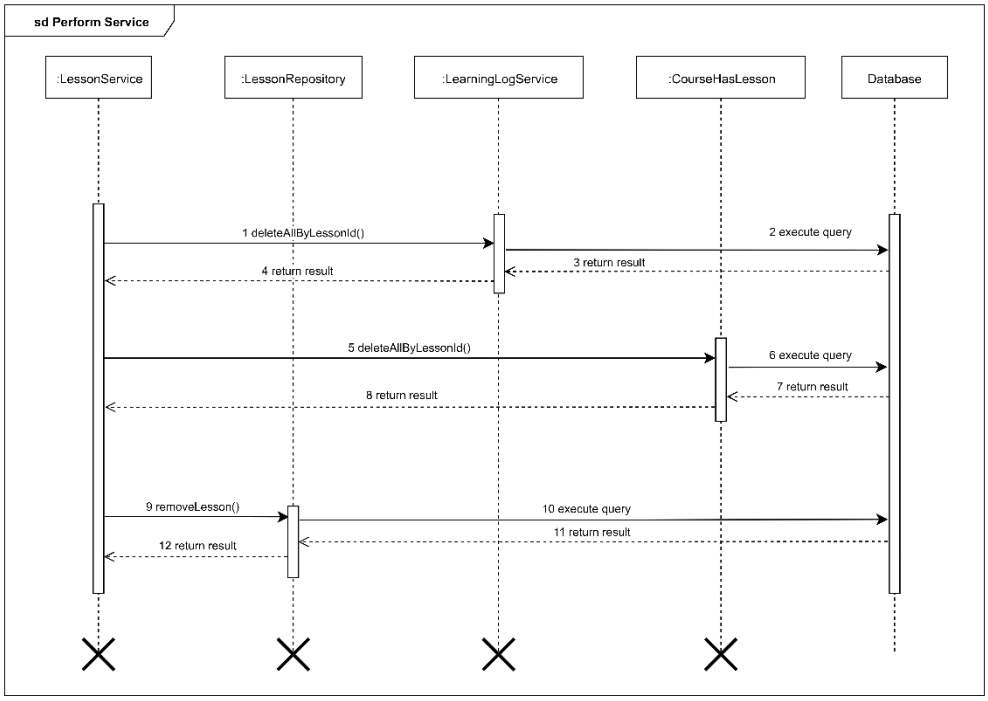
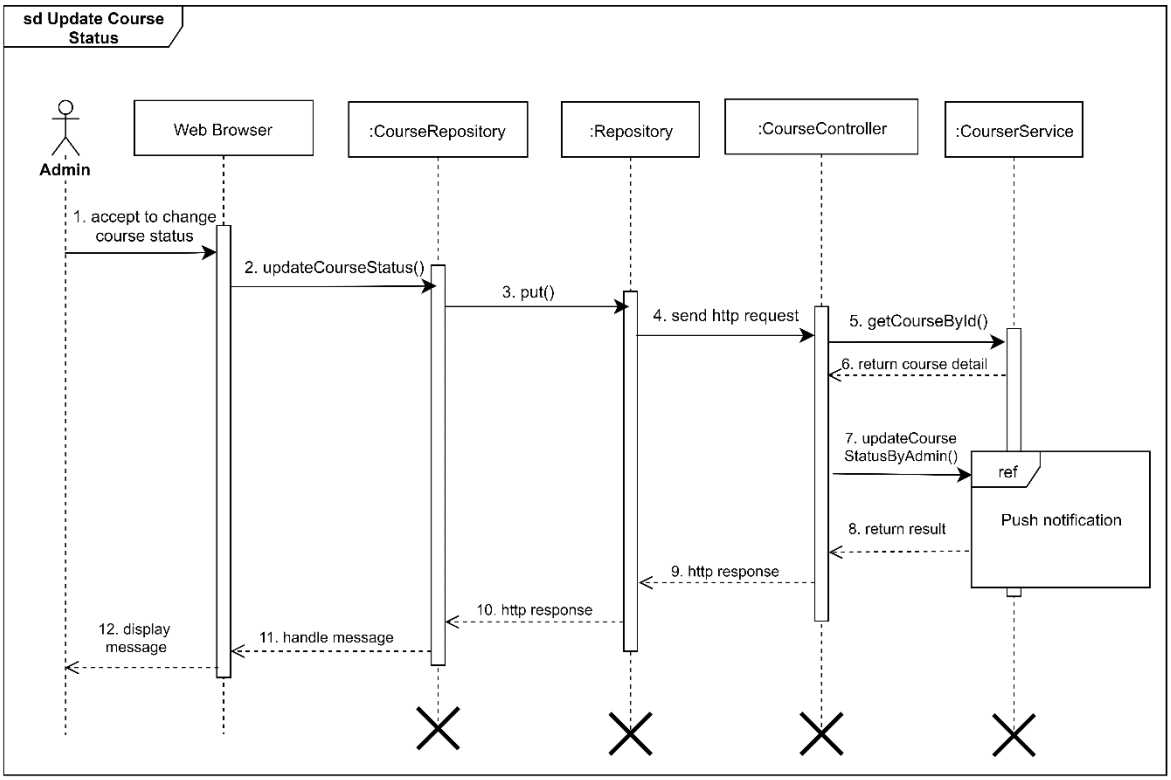


Figure 68: Sequence Diagram - <Instructor> Remove Lesson

4.4.5 Admin

4.4.5.1 Update Course Status

Summary: this diagram shows process of update course status. This process is used for approve, reject or unpublish course.



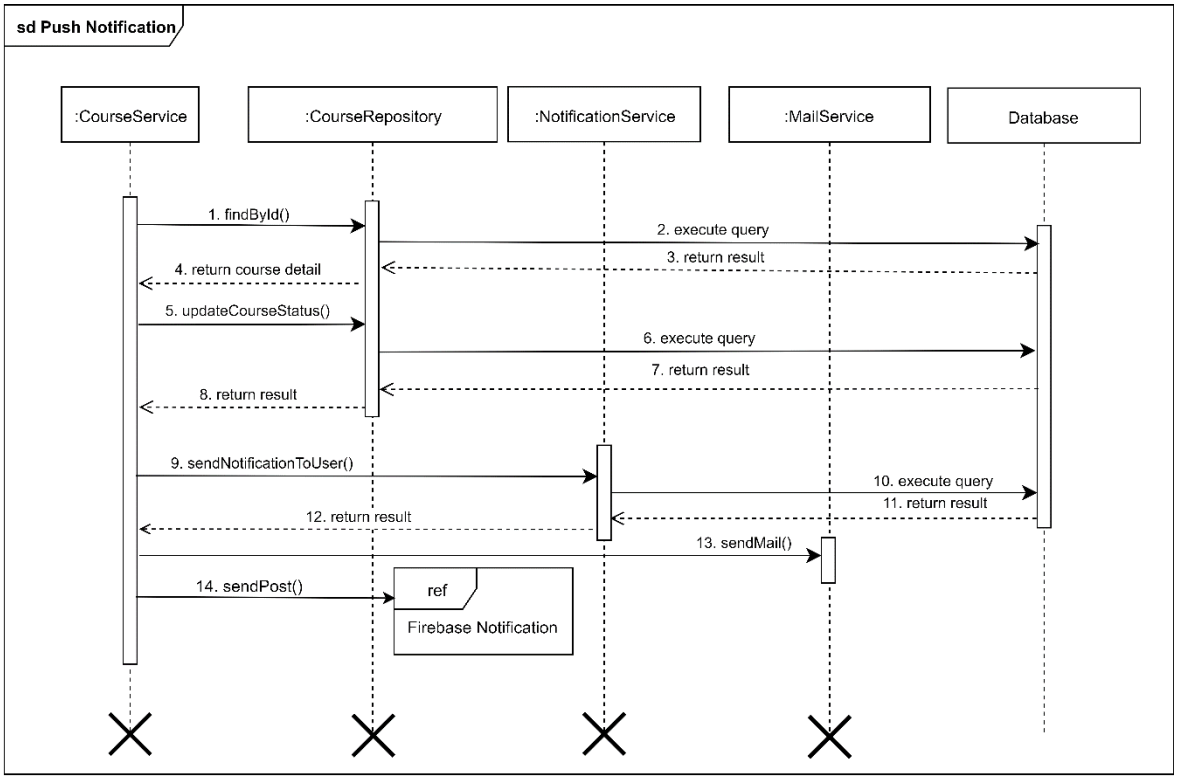


Figure 69: Sequence Diagram - <Admin> Update Course Status

4.4.6 System Bot (Play Chess With Learner)

Summary: this diagram shows process of play chess with learner.

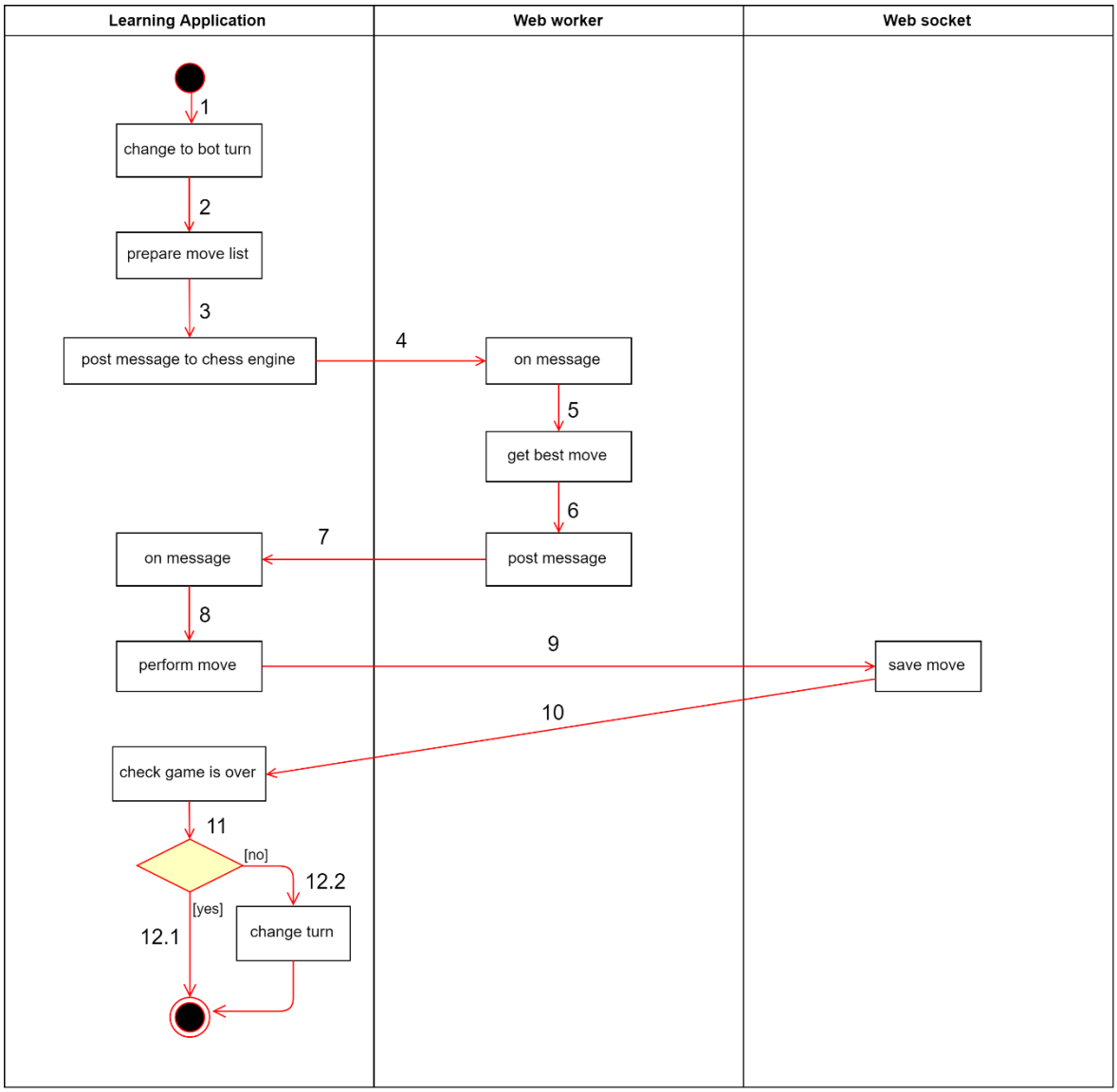


Figure 70: Activity Diagram - <System Bot> Play Chess With Learner

5. User Interface Design

5.1 Learner Interface Design

5.1.1 Home Page

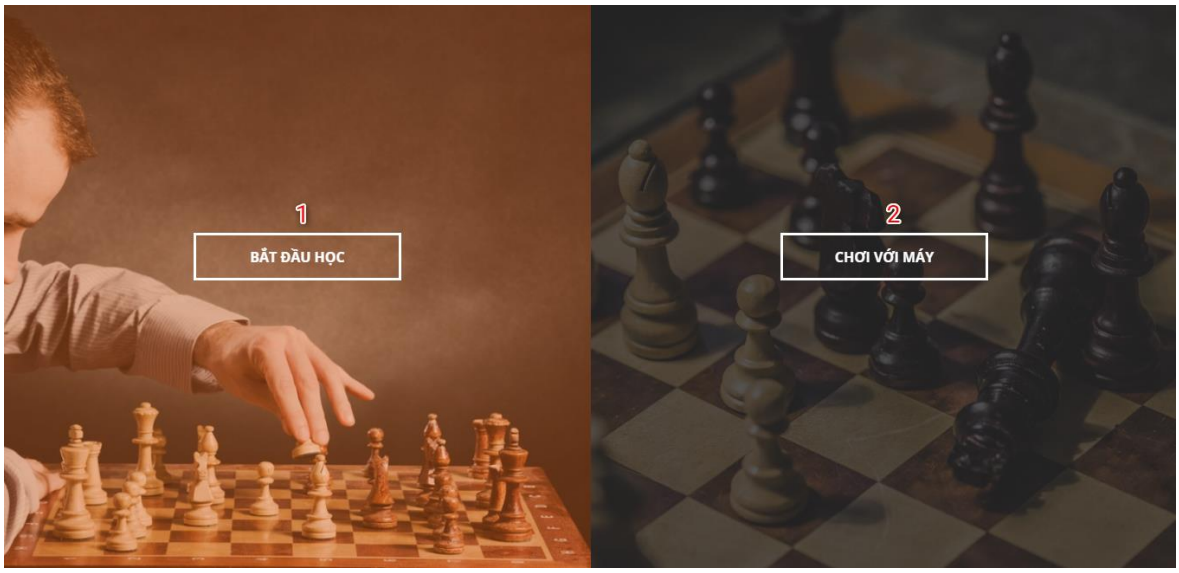


Figure 71: Home Page

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
1	Learning	Choose learning chess	N/A	Redirects to courses page
2	Playing	Choose playing chess	N/A	Redirects to playing chess page

Table 73: Home Page –Buttons/Hyperlinks

5.1.2 View Published Course

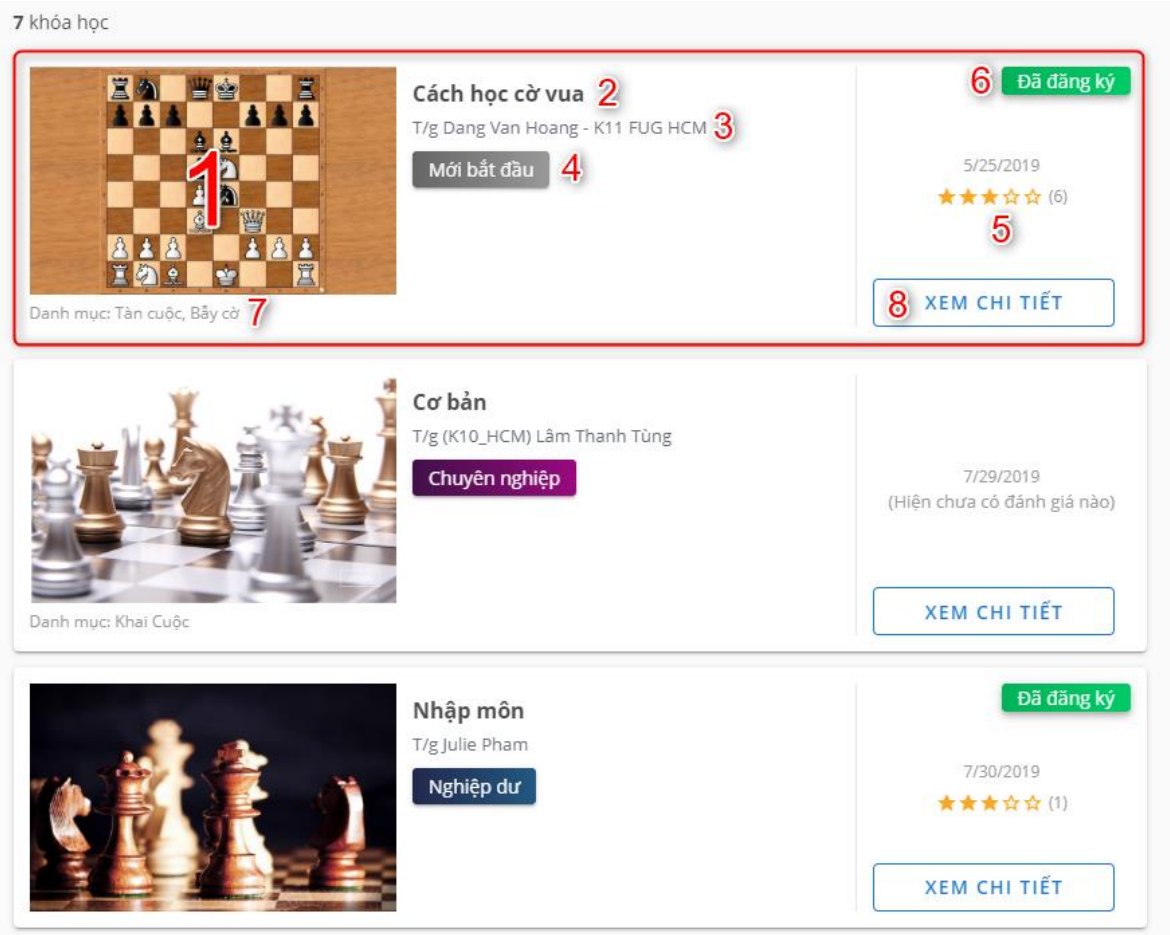


Figure 72: View Published Course

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	courseImg	Course's image	Yes	Yes	Image	N/A	N/A
2	courseName	Course's name	Yes	Yes	Text	N/A	N/A
3	courseAuthor	Course's author	Yes	Yes	Text	N/A	N/A
4	requiredLevel	Course's required level	Yes	Yes	Label	N/A	N/A
5	courseRating	Course's rating	Yes	Yes	Text	N/A	N/A
6	isEnrolled	Mark course is enrolled	Yes	Yes	Label	N/A	N/A

7	courseCate	Course's category	Yes	Yes	Text	N/A	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**		**Validation**	**Outcome**		
11	btnViewCourse	View course details		N/A	System redirects to course details page		

Table 74: View Published Course –Fields and Buttons/Hyperlinks

5.1.3 View course details

5.1.3.1 Before Enrol

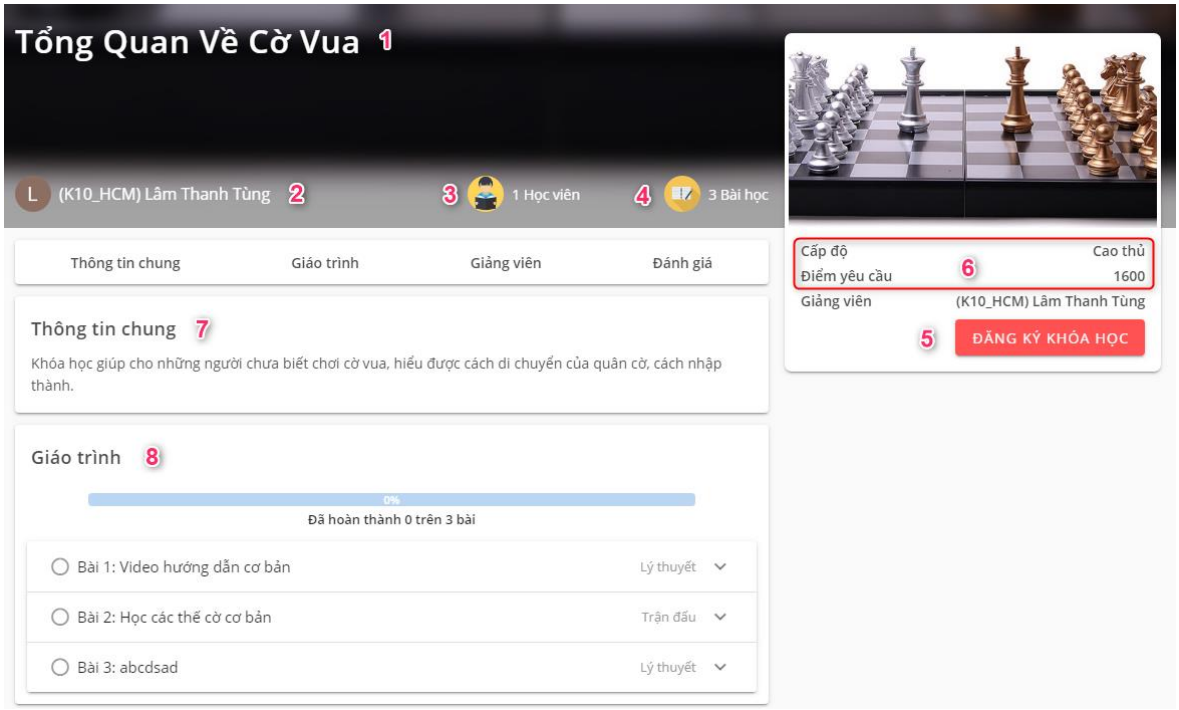


Figure 73: Before Enrol

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	courseName	Course'name	Yes	Yes	Text	N/A	N/A
2	instructorName	Course's author name	Yes	Yes	Text	N/A	N/A
3	learnerNumber	Enrol count	Yes	Yes	Text	N/A	N/A
4	lessonNumber	Number of lesson in course	Yes	Yes	Text	N/A	N/A

6	pointInfo	Course's point information	Yes	Yes	Text	N/A	N/A
7	description	Course's description	Yes	Yes	Text	N/A	N/A
8	lessonList	List of lesson	Yes	Yes	Panel	N/A	N/A
9	lbLessonType	Lesson type name	Yes	Yes	Text	N/A	N/A
10	icIsLearned	Mark learned lesson	Yes	Yes	Icon	N/A	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**	**Validation**	**Outcome**			
11	btnEnrol	Enrol to course	N/A	Course can be enrolled			

Table 75: Before Enrol – Fields and Buttons/Hyperlinks

5.1.3.2 After Enrol

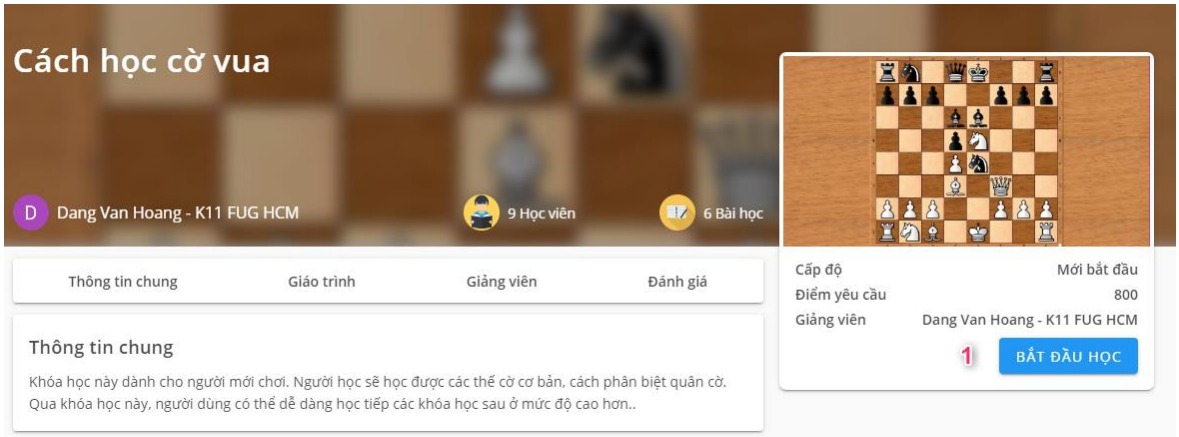


Figure 74: After Enrol

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
1	btnStart	Start to learn course	N/A	System redirects to lesson page

Table 76: After Enrol – Buttons/Hyperlinks

5.1.3.3 Review Course

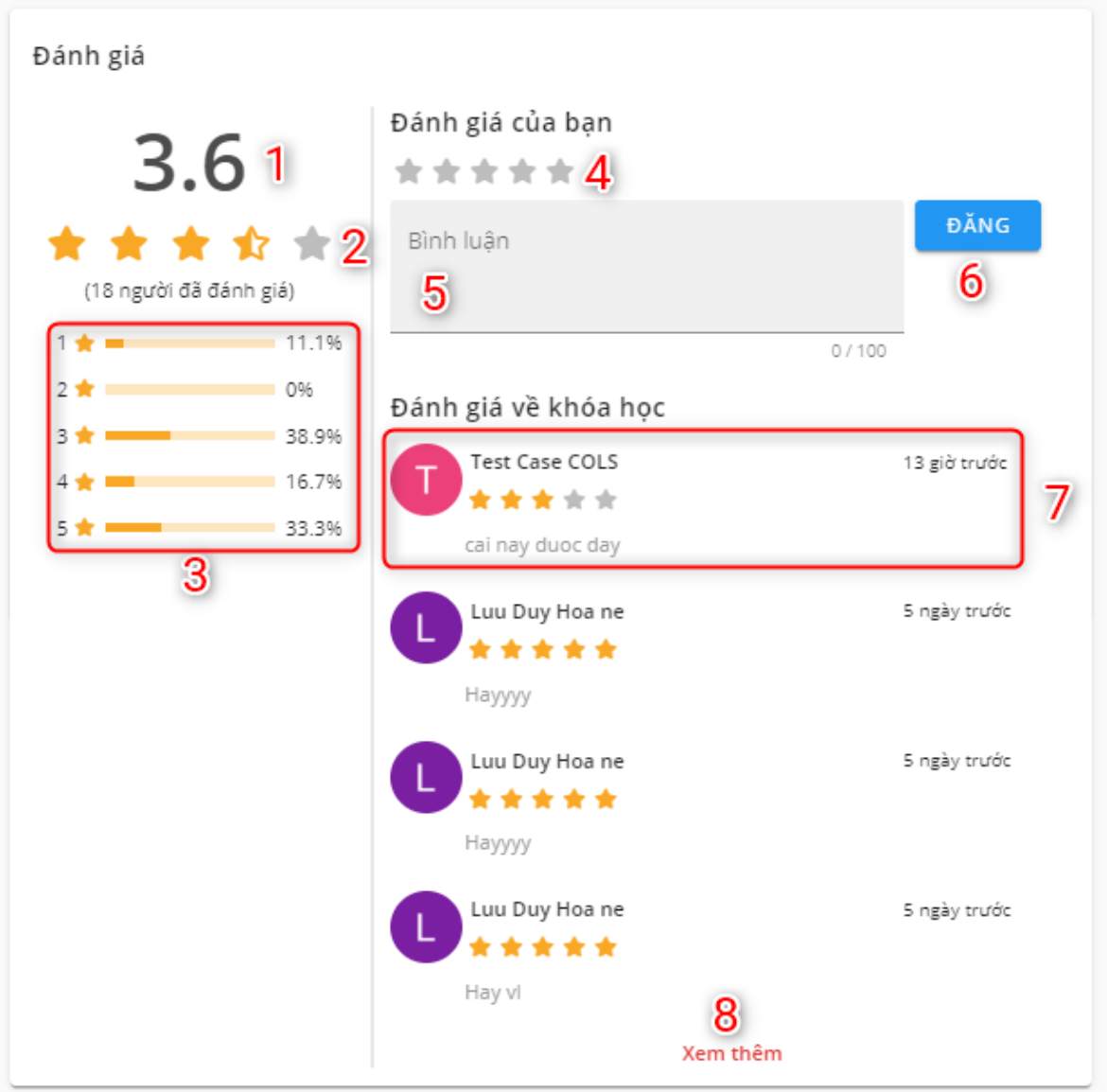


Figure 75: Review Course

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	avgRating	Average of rating point	Yes	Yes	Text	N/A	N/A
2	avgRatingBy Star	Rating point with star	Yes	Yes	Rating	N/A	N/A
3	ratingPointP ercentage	Percentage for each rating point	Yes	Yes	Text	N/A	N/A
5	txtReview	Review input field	No	Yes	Textarea	String	N/A

7	sampleReview	Sample succeed review	Yes	Yes	Box	N/A	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**		**Validation**	**Outcome**		
4	btnRating	Rating button for learner		N/A	N/A		
6	btnSubmit	Review's submit button		N/A	System checks and updates course ratings		
8	btnMore	Review's view more button		N/A	System load more reviews		

Table 77: Review Course – Fields and Buttons/Hyperlinks

5.1.3.4 Lesson List

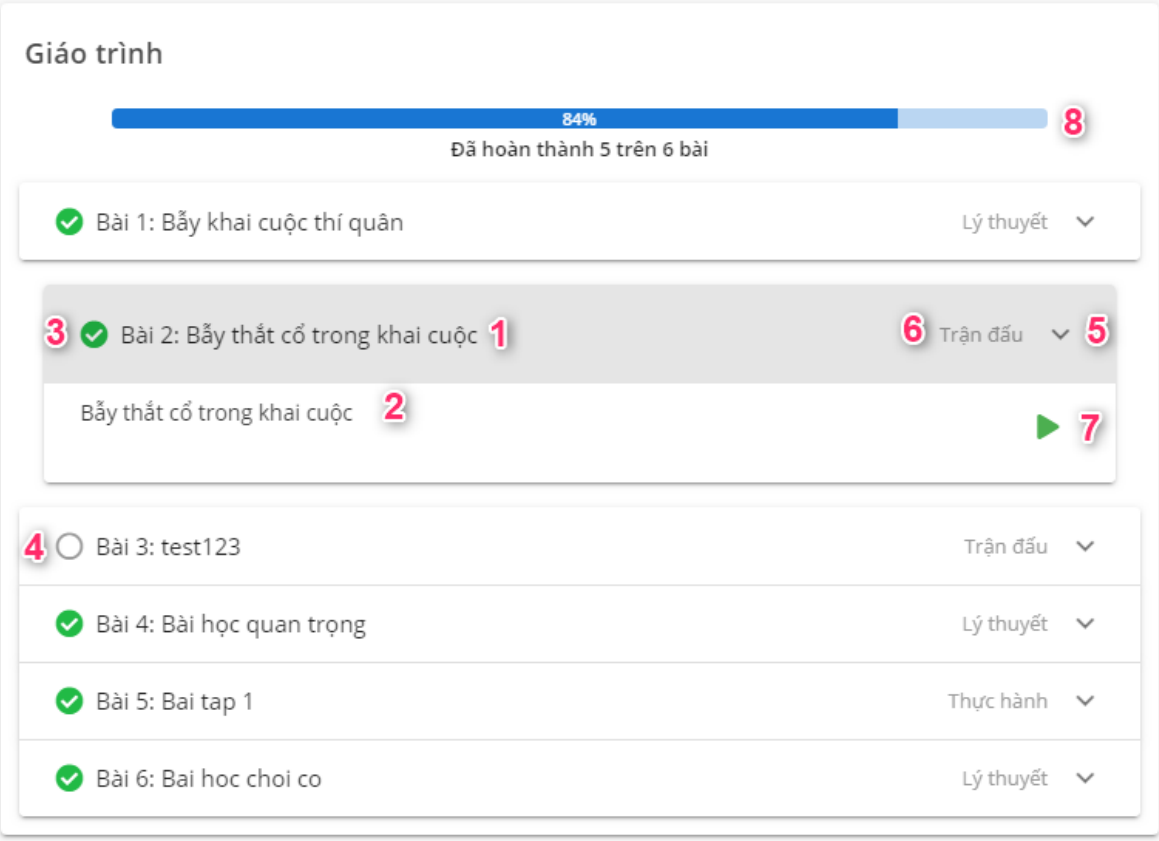


Figure 76: Lesson List

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	lessonName	Lesson's name	Yes	Yes	Text	N/A	N/A
2	lessonDescription	Lesson's description	Yes	Yes	Text	N/A	N/A

3	finishedIcon	Mark lesson is finished	Yes	Yes	Icon	N/A	N/A
4	unfinishedIcon	Lesson has not be finished	Yes	Yes	Icon	N/A	N/A
6	lessonType	Lesson's type	Yes	Yes	Text	N/A	N/A
8	learningProgress	Lesson's progress	Yes	Yes	Progress 's bar	N/A	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**		**Validation**	**Outcome**		
5	btnDropDown	Show lesson description button		N/A	System displays lesson information		
7	btnLearnLeson	Learn a specific leson button		N/A	System redirects to a specific lesson		

Table 78: Lesson List – Fields and Buttons/Hyperlinks

5.1.4 Learn Interactive Lesson

5.1.4.1 Main UI

Figure 77: Learn Interactive Lesson

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	chessboard	Chessboard	Yes	Yes	Chessboard	N/A	N/A

2	moveBox	Move container	No	Yes	Card	N/A	N/A
3	moveDirectionButtonGroup	Move direction container	No	Yes	Button group	N/A	N/A
6	moveContent	Move content container	Yes	Yes	Card	N/A	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**	**Validation**	**Outcome**			
2	move	A move in chess game	N/A	Show chess position corresponding with move			
4	btnPreviousLesson	Change to previous lesson	N/A	Show previous lesson			
5	btnNextLesson	Change to next lesson	N/A	Show next lesson			

Table 79: Learn Interactive Lesson – Fields and Buttons/Hyperlinks

5.1.4.2 Move Container

Figure 78: Move Container

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	moveIndex	Index of move	Yes	Yes	Text	N/A	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**	**Validation**	**Outcome**			

2	mormalMove	A move in chess game	N/A	Show chess position corresponding with move
3	currentMove	Current selected move with current pieces position on Chessboard	N/A	N/A
4	depth2Move	A move in depth 2	N/A	Show chess position corresponding with move
5	depth3Move	A move in depth 3	N/A	Show chess position corresponding with move
6	btnFirstMove	Turn to first move button	N/A	Show first position
7	btnPreviousMove	Turn to previous move button	N/A	Show previous position
8	btnNextMove	Turn to next move button	N/A	Show next position
9	btnLastMove	Turn to last move button	N/A	Show last position

Table 80: Move Container – Fields and Buttons/Hyperlinks

5.1.5 Learn Uninteractive Lesson

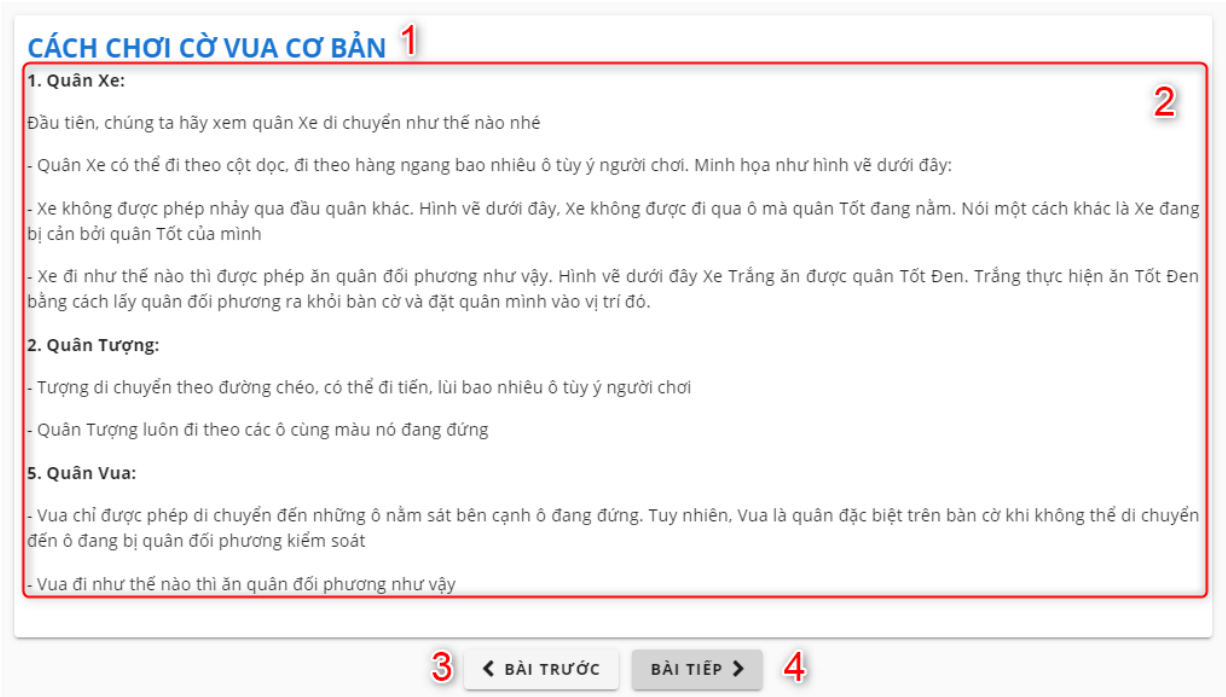


Figure 79: Learn Uninteractive Lesson

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**

1	lessonName	Lesson's name	Yes	Yes	Text	N/A	N/A
2	lessonContent	Lesson's content	Yes	Yes	Text	N/A	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**		**Validation**	**Outcome**		
3	btnPreviousLesson	Change to previous lesson		N/A	Show previous lesson		
4	btnNextLesson	Change to next lesson		N/A	Show next lesson		

Table 81: Learn Uninteractive Lesson – Fields and Buttons/Hyperlinks

5.1.6 Do Exercise

5.1.6.1 Main UI

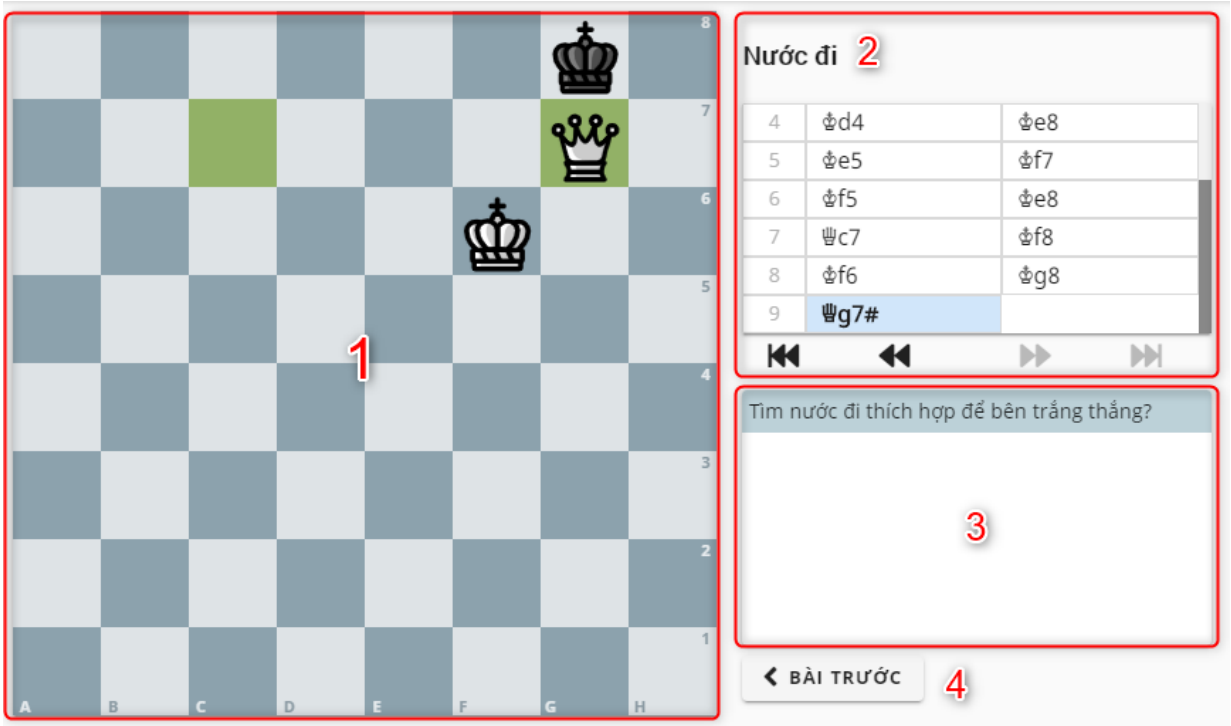


Figure 80: Do Exercise

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	chessboard	Chessboard	Yes	Yes	Chessboard	N/A	N/A
6	moveBox	Move container	No	Yes	Card	N/A	N/A
7	exerciseInfo	Exercise's information	Yes	Yes	Card	N/A	N/A
BUTTONS/HYPERLINKS							

No	Function	Description	Validation	Outcome
4	btnPreviousLesson	Change to previous lesson	N/A	Show previous lesson

Table 82: Do Exercise – Fields and Buttons/Hyperlinks

5.1.6.2 Finish Exercise

Figure 81: Finish Exercise

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	result	Exercise's result	Yes	Yes	Text	N/A	N/A

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
2	btnOK	Close dialog	N/A	Show next lesson

Table 83: Finish Exercise – Fields and Buttons/Hyperlinks

5.1.7 Play Chess

5.1.7.1 Start

Figure 82: Play Chess

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
1	btnStart	Start to play chess	N/A	Show Control Dialog

Table 84: Play Chess – Buttons/Hyperlinks

5.1.7.2 Game control

Figure 83: Game control

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
3	txtGameTime	Game's time	No	Yes	Text	Time	N/A

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
1	sdGameLevel	Pick game's level	N/A	N/A
2	btnColor	Pick game's color	N/A	N/A
4	btnClose	Close Game control dialog	N/A	N/A
5	btnStart	Start game	N/A	N/A

Table 85: Game Control – Fields and Buttons/Hyperlinks

5.1.7.3 Playing

Figure 84: Playing

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	chessboard	Chessboard	Yes	Yes	Chessboard	N/A	N/A
2	winPoint	Bonus point when learner win	Yes	Yes	Text	N/A	N/A
3	learnerPoint	Current point of learner	Yes	Yes	Text	N/A	N/A
4	botClock	Playing time for bot	Yes	Yes	Text	N/A	N/A
5	learnerClock	Playing time for learner	Yes	Yes	Text	N/A	N/A
6	moveBox	Move container	No	Yes	Card	N/A	N/A
7	gameInfo	Game's information	Yes	Yes	Card	N/A	N/A

Table 86: Playing - Fields

5.1.8 Profile

5.1.8.1 Account Information

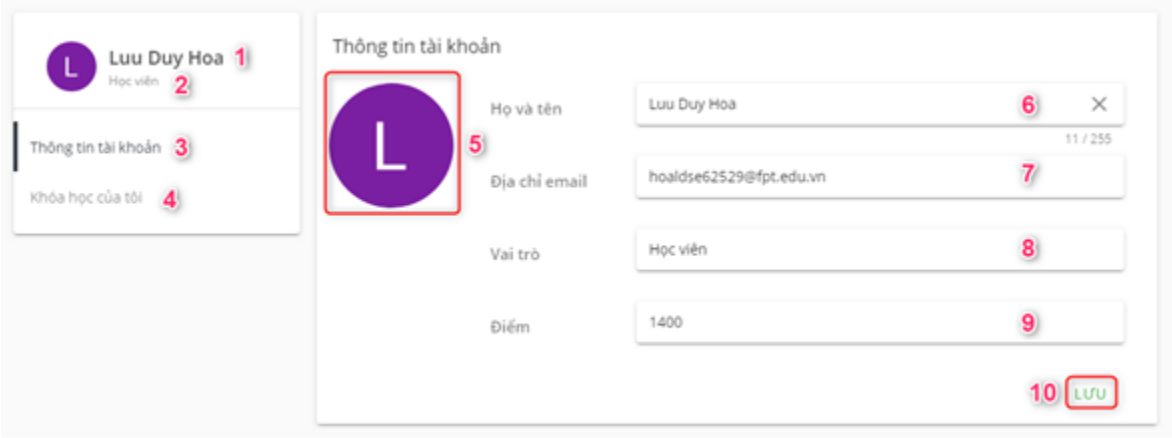


Figure 85: Play Chess

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
6	txtFullname	User's fullname	No	Yes	Text	N/A	N/A
7	txtEmail	User' email	Yes	Yes	Text	N/A	N/A
8	txtRole	User' role	Yes	Yes	Text	N/A	N/A
9	txtPoint	User' point	Yes	Yes	Text	N/A	N/A

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
3	accountInformationTab	Change to account information tab	N/A	System shows information tab
4	enrolledCourseTab	Change to enrolled course tab	N/A	System shows enrolled course tab
5	btnAvatar	Change avatar dialog	N/A	System show change avatar dialog
10	btnSave	Save profile button	N/A	Saving profile

Table 87: Account Information – Fields and Buttons/Hyperlinks

5.1.8.2 Enrolled course

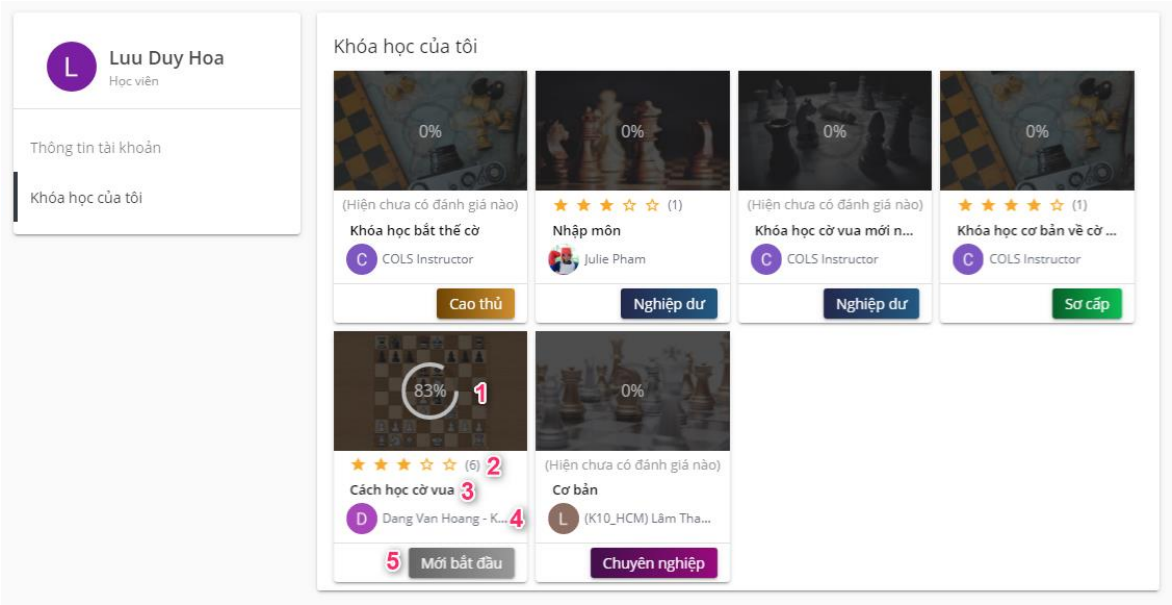


Figure 86: Game control

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	progressBar	User's learning progress	Yes	Yes	Progress bar	N/A	N/A
2	avgRating	Course's rating	Yes	Yes	Rating	N/A	N/A
3	courseTitle	Course's title	Yes	Yes	Text	N/A	N/A
4	authorName	Course's author	Yes	Yes	Text	N/A	N/A
5	courseLevel	Course's level	Yes	Yes	Text	N/A	N/A

Table 88: Enrolled Course - Fields

5.2 Instructor Interface Design

5.2.1 Menu

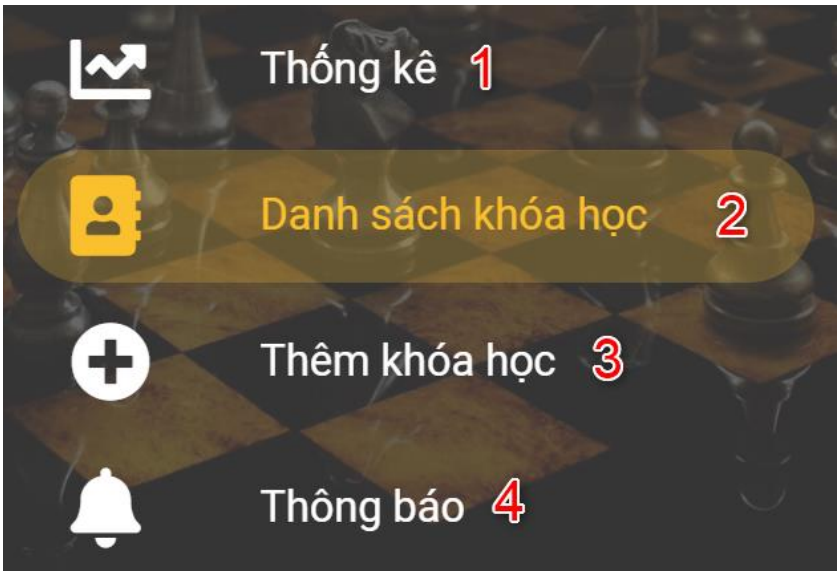


Figure 87: Instructor Menu

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
1	btnStatistic	Redirect to Statistic Page	N/A	Change to Statistic Page
2	btnCourseList	Redirect to Course List Page	N/A	Change to Course List Page
3	btnCreateCourse	Redirect to Create course page	N/A	Change to Create course Page
4	btnNotification	Redirect to Notification page	N/A	Change to Notification Page

Table 89: Instructor Menu –Buttons/Hyperlinks

5.2.2 Create course

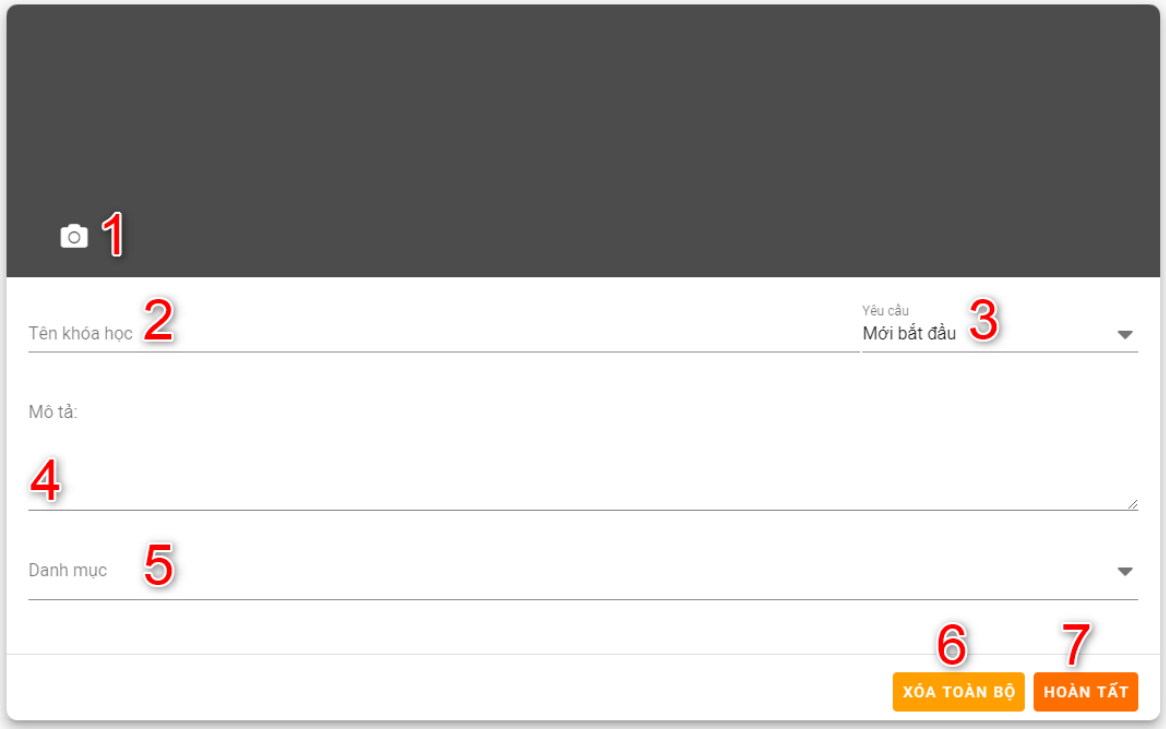


Figure 88: Create course

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	txtCourseImage	Course's image	No	Yes	Text	File	N/A
2	txtCourseName	Course'sname	No	Yes	Text	Text	N/A
3	cmbRequiredLevel	Course's required level	No	Yes	Combobox	N/A	N/A
4	txtDescription	Course's description	No	Yes	Text area	Text	N/A
5	cmbCategory	Course's category	No	Yes	Combobox	N/A	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**	**Validation**	**Outcome**			
6	btnReset	Reset form	N/A	Remove all data			
7	btnCreate	Create a new course	N/A	Change to new course details page			

Table 90: Create course – Fields and Buttons/Hyperlinks

5.2.3 Course List

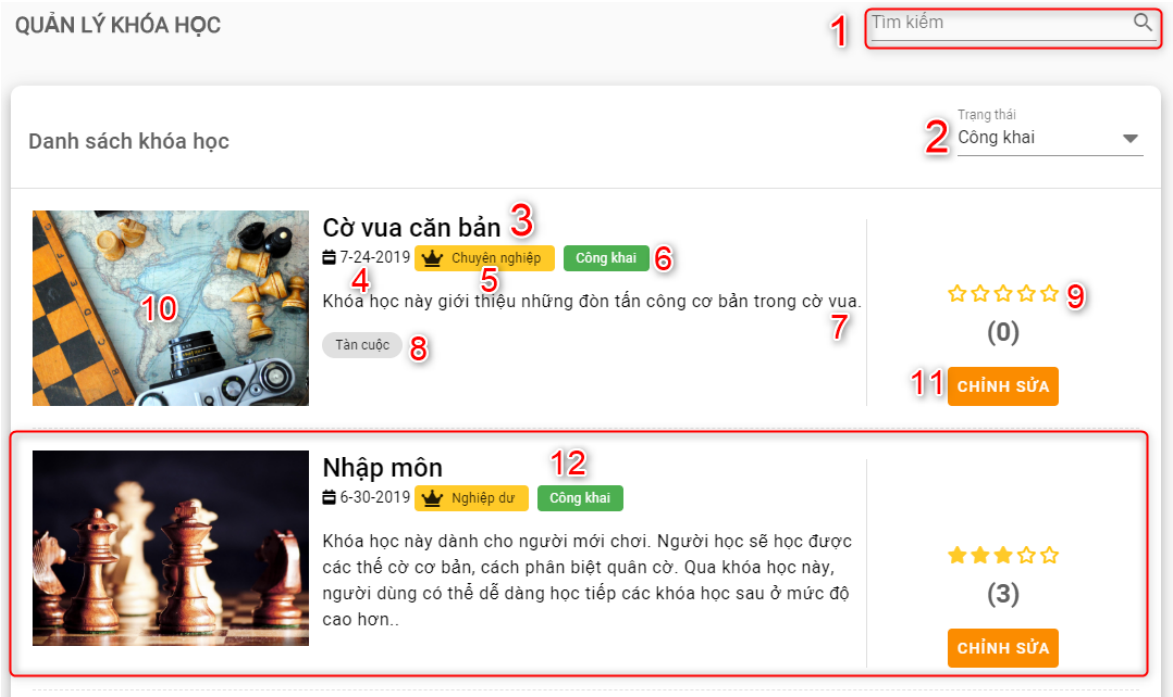


Figure 89: Course List

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
2	courseFilterStatus	Filter course by status	Yes	Yes	Select box	List	N/A
3	courseName	Course's name	Yes	Yes	Text	String	N/A
4	courseCreatedDate	Course's create date	Yes	Yes	Text	String	N/A
5	courseRequiredLevel	Course's required Level	Yes	Yes	Text	String	N/A
6	courseStatus	Course's status	Yes	Yes	Text	String	N/A
7	courseDescription	Course'sDescription	Yes	Yes	Text	String	N/A
8	listCategoryByCourse	Course's Category	Yes	Yes	Text	String	N/A
9	courseRating	Course's rating	Yes	Yes	Rating	Number	N/A

10	courseImage	Course's image	Yes	Yes	Image	File	N/A
11	courseItem	Course item in list	Yes	Yes	Panel	N/A	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**		**Validation**	**Outcome**		
1	search	search course		N/A	Show matching course		
11	gotoCourseDetail	go to course detail		N/A	redirect to course detail page		

Table 91: Course List – Fields and Buttons/Hyperlinks

5.2.4 Manage Course

5.2.4.1 Overview

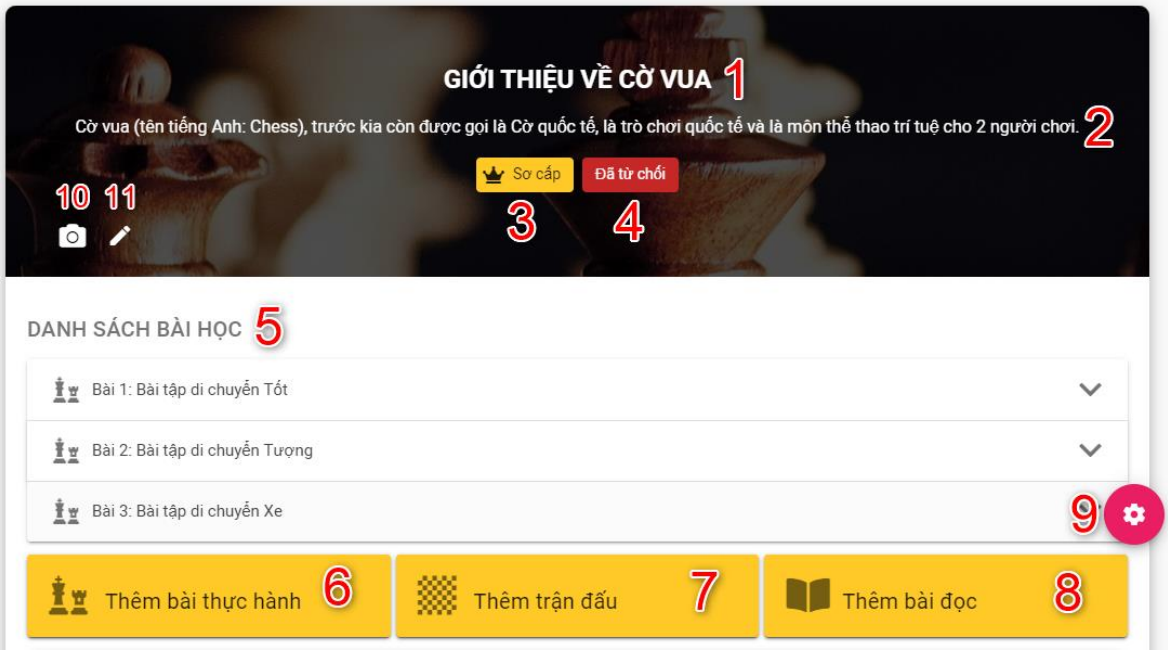


Figure 90: Course Overview

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	courseName	Course's name	Yes	Yes	Text	String	N/A
2	courseDescription	Course'sDescription	Yes	Yes	Text	String	N/A
3	courseRequiredLevel	Course's required level	Yes	Yes	Text	String	N/A

4	courseStatus	Course's status	Yes	Yes	Text	String	N/A
9	listLesson	Course's list lesson	Yes	Yes	Panel	List	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**		**Validation**	**Outcome**		
6	openLesson Dialog	open interactive lesson dialog		N/A	Show interactive lesson dialog		
7	openLesson Dialog	open exercise dialog		N/A	Show exercise dialog		
8	openLesson Dialog	open un-interactive lesson dialog		N/A	Show un-interactive lesson dialog		
9	actionSheet	open management course status		N/A	Show course status management		
10	courseImage	open course image input field		N/A	Show file input dialog		
11	editedCourse	open edit course details dialog		N/A	Show edited course details dialog		

Table 92: Course Overview – Fields and Buttons/Hyperlinks

5.2.4.2 Lesson List

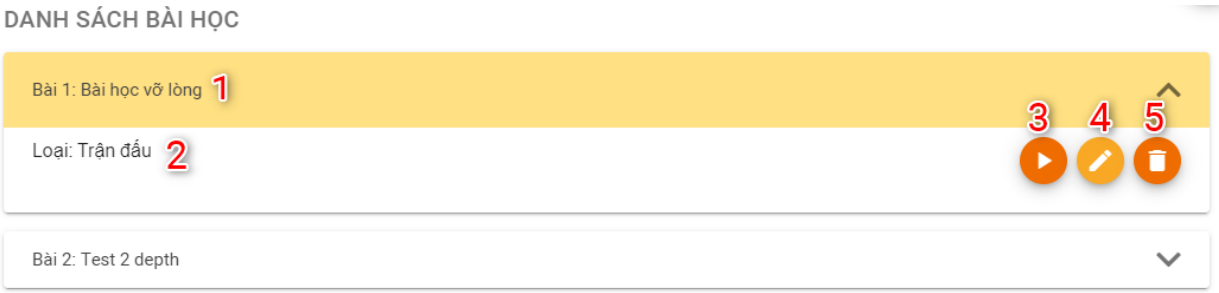


Figure 91: Lesson List

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	courseName	Course's name	Yes	Yes	Text	String	N/A
2	courseStatus	Course's status	Yes	Yes	Text	String	N/A

Table 93: Lesson List - Fields

5.2.4.3 Settings

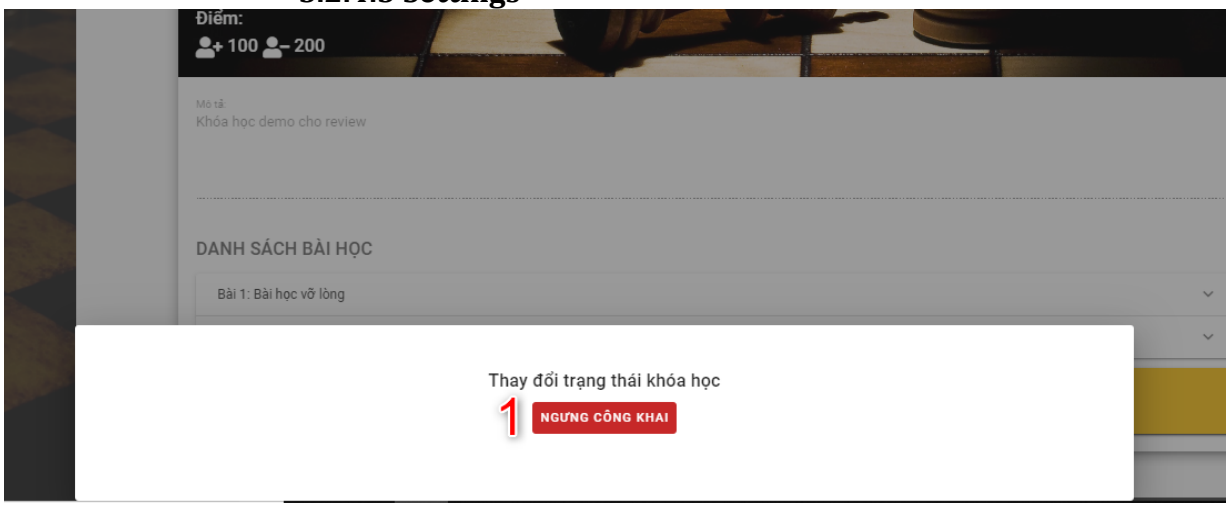


Figure 92: Settings

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
1	changeCourseStatus	Change course status	N/A	Change course status

Table 94: Setiings – Buttons/Hyperlinks

5.2.5 Create Lesson

5.2.5.1 Add Interactive Lesson

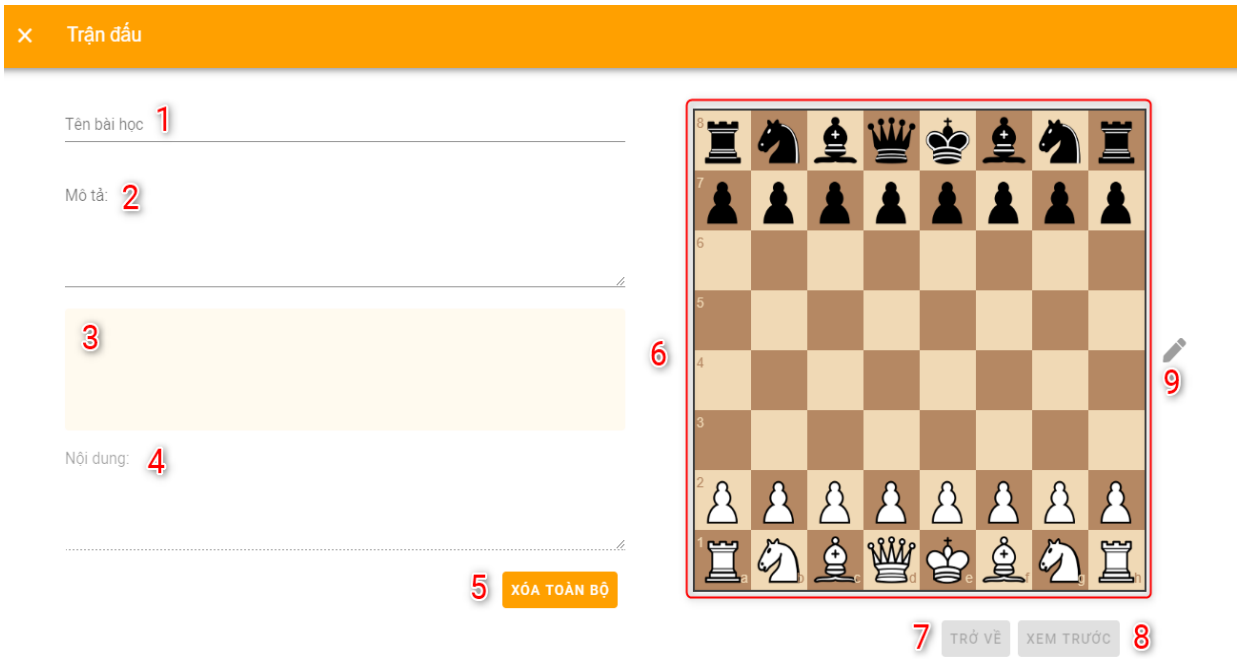


Figure 93: Add Interactive Lesson

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**

1	lessonName	Lesson's name	Yes	Yes	Textbox	String	N/A
2	lessonDescription	Lesson's description	Yes	Yes	Textbox	String	N/A
3	lessonFen	Lesson's fen	Yes	Yes	Text	String	N/A
4	lessonContent	Lesson's content	Yes	Yes	Textbox	String	N/A
6	chessboard	Lesson's chessboard	Yes	Yes	chessboard	N/A	N/A

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
5	resetForm	clear all fields in form	N/A	clear all fields in form
7	closeDialog	close interactive lesson dialog	N/A	Close interactive lesson dialog
8	preview	preview lesson	N/A	Show preview lesson
9	editChessboard	edit chess board	N/A	Edit chessboard

Table 95: Add Interactive Lesson – Fields and Buttons/Hyperlinks

5.2.5.2 Add Uninteractive Lesson

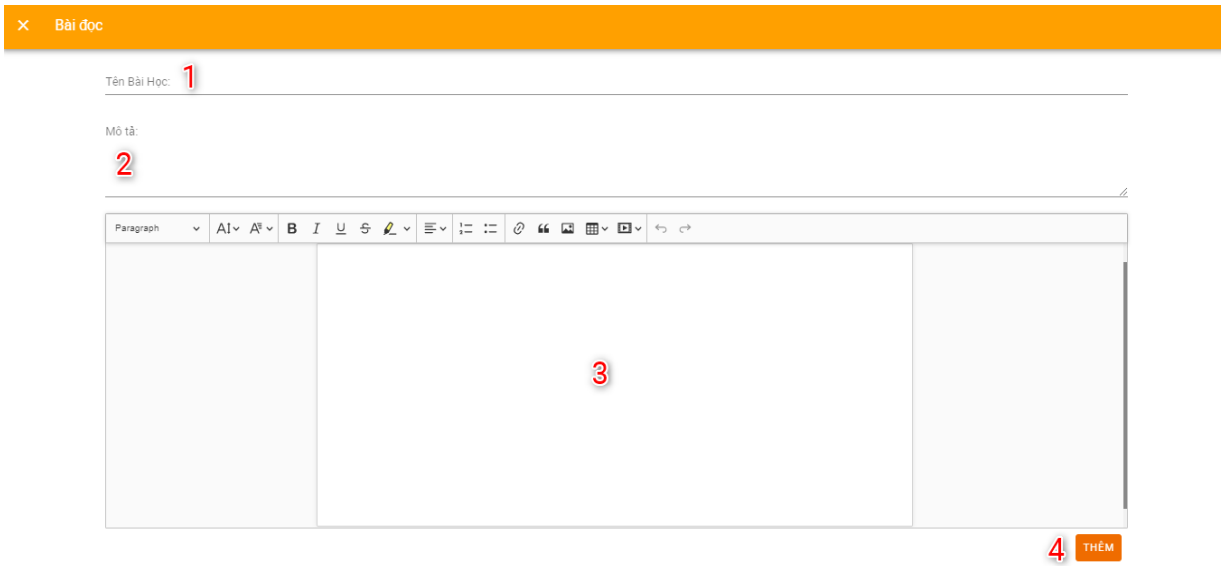


Figure 94: Add Uninteractive lesson

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	lessonName	Lesson's name	Yes	Yes	Textbox	String	N/A

2	lessonDescription	Lesson's description	Yes	Yes	Textbox	String	N/A
3	lessonContent	Lesson's content	Yes	Yes	Textbox	String	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**		**Validation**	**Outcome**		
4	createUninteractiveLesson	Create an un-interactive lesson		N/A	Create an un-interactive lesson		

Table 96: Add Uninteractive Lesson – Fields and Buttons/Hyperlinks

5.2.5.3 Add Exercise

Figure 95: Add Exercise

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	exerciseName	Exercise's name	Yes	Yes	Textbox	String	N/A
2	exerciseQuesion	Exercise's Question	Yes	Yes	Textbox	String	N/A
3	exerciseFen	Exercise's fen	Yes	Yes	Text	String	N/A
4	listAnswer	Exercise's list answer	Yes	Yes	Box	List	N/A
5	exerciseAnswerRightContent	Exercise's answer right content	Yes	Yes	Textbox	String	N/A

6	exerciseAnswerWrongContent	Exercise's answer wrong content	Yes	Yes	Textbox	String	N/A
9	chessboard	Lesson's chessboard	Yes	Yes	chessboard	N/A	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**	**Validation**	**Outcome**			
8	resetForm	clear all fields in form	N/A	clear all fields in form			
11	closeDialog	close interactive lesson dialog	N/A	Close interactive lesson dialog			
12	prepareExercise	Create exercise	N/A	Create exercise			

Table 97: Add Exercise – Fields and Buttons/Hyperlinks

5.2.5.4 Dynamic Chess Position

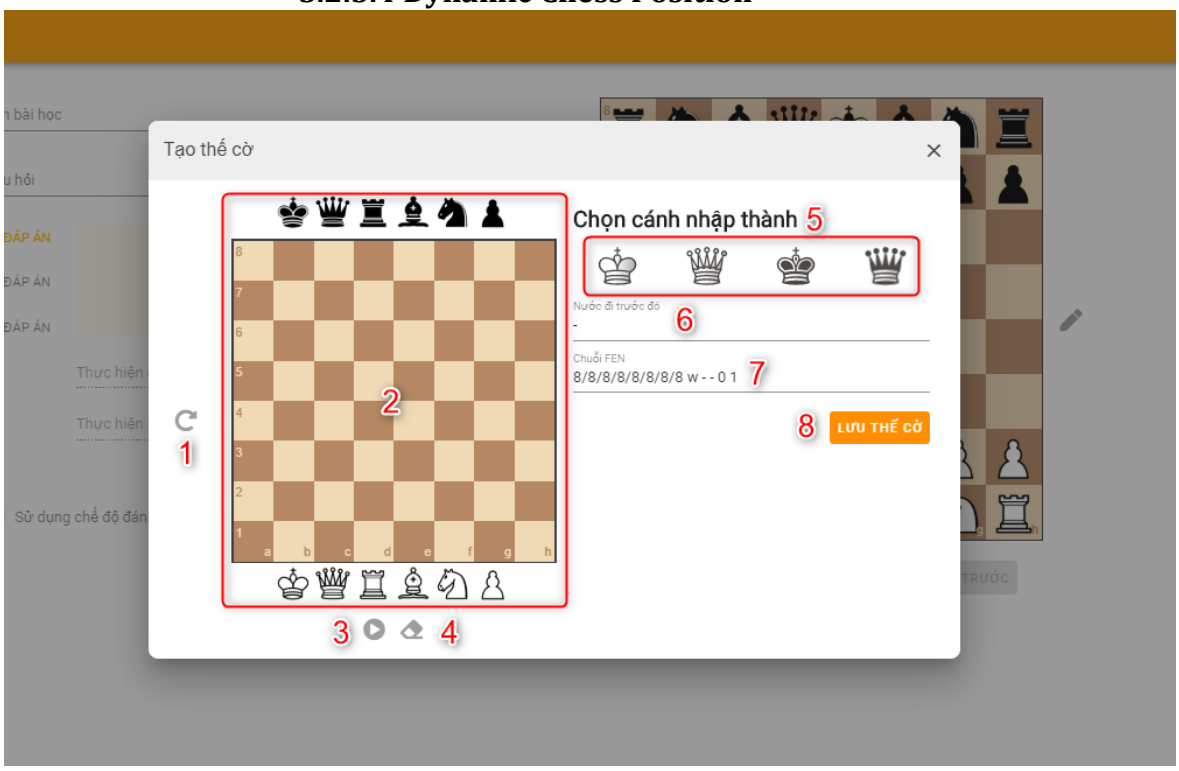


Figure 96: Dynamic Chess Position

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
2	chessboard	Chessboard	Yes	Yes	chessboard	N/A	N/A

6	chessboardPreviousMove	Chessboard's previous move	Yes	Yes	Text	String	N/A
7	chessboardFen	Chessboard's fen	Yes	Yes	Text	String	N/A
BUTTONS/HYPERLINKS							
No	**Function**	**Description**		**Validation**	**Outcome**		
1	flipBoard	Swap chess		N/A	Swap chess		
3	startChess	Create default chessboard		N/A	Create default chessboard		
4	clearBoard	Clear all chess in chessboard		N/A	Clear all chess in chessboard		
5	chooseChess	Choose chess		N/A	Choose chess		
8	saveFen	Save fen		N/A	Save fen		

Table 98: Dynamic Chess position – Fields and Buttons/Hyperlinks

5.2.5.5 Preview Lesson

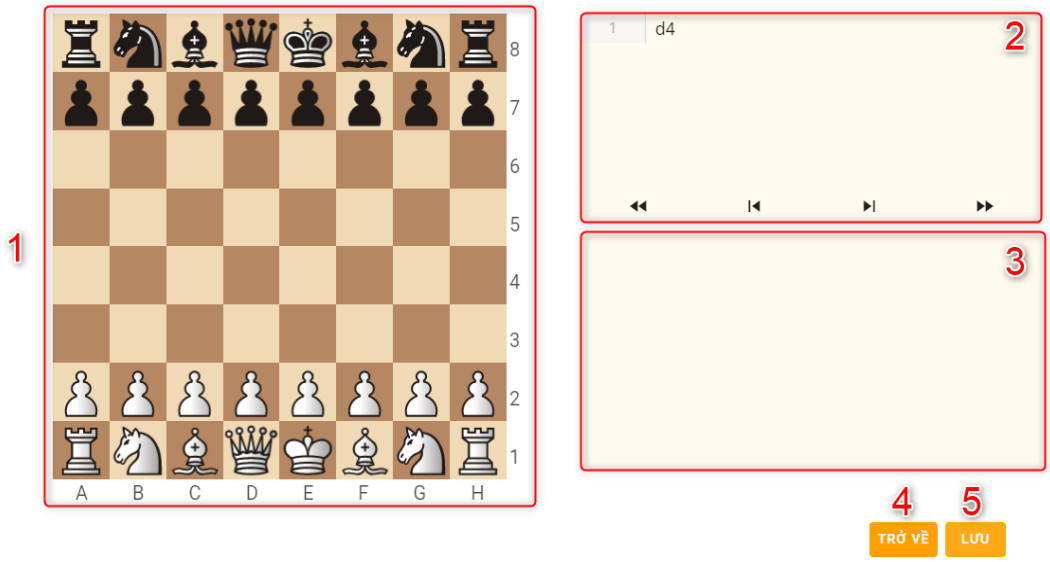


Figure 97: Preview Lesson

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	chessboard	chessboard	Yes	Yes	chessboard	N/A	N/A
2	moveBox	Move container	No	Yes	Card	N/A	N/A
3	gameInfo	Game's information	Yes	Yes	Card	N/A	N/A

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
4	closeDialog	close interactive lesson dialog	N/A	Close interactive lesson dialog
5	saveFen	Save Fen	N/A	Save Fen

Table 99: Preview Lesson – Fields and Buttons/Hyperlinks

5.3 Admin Interface Design

5.3.1 Create Category

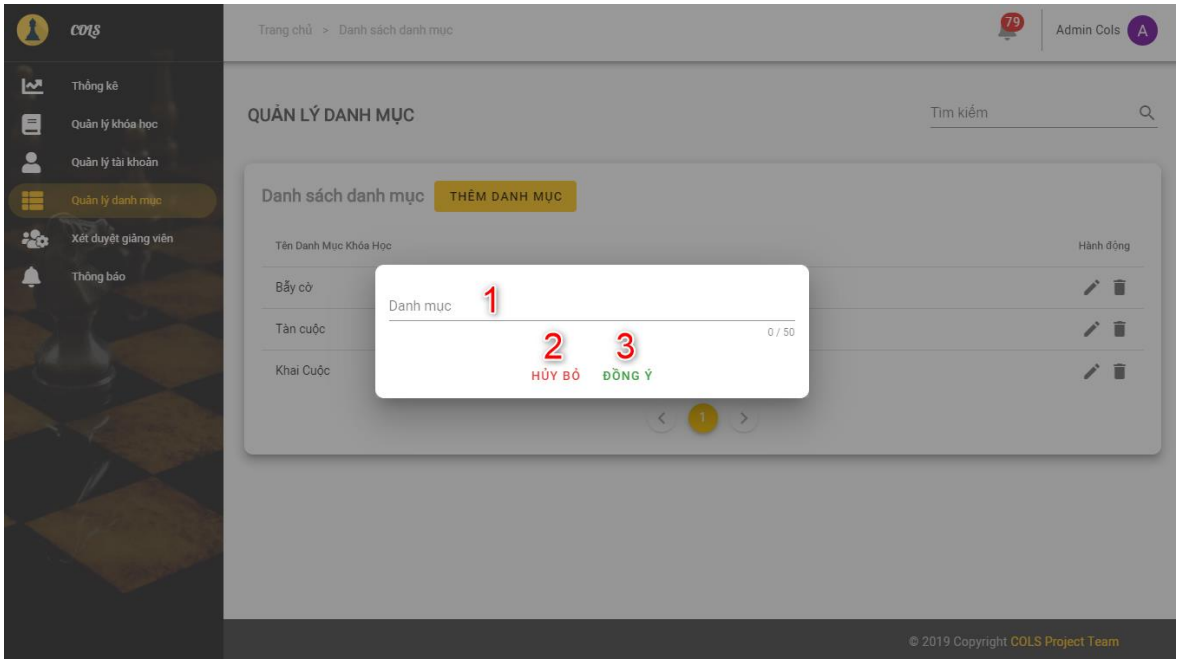


Figure 98: Create Category

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	categoryName	Category's name	Yes	Yes	Text	Field	N/A

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
2	btnClose	Close category form button	N/A	System closes created category form
5	btnSave	Create category button	N/A	System creates a category

Table 100: Create Category –Fields and Buttons/Hyperlinks

5.3.2 Review Instructor

5.3.2.1 List of New Instructor

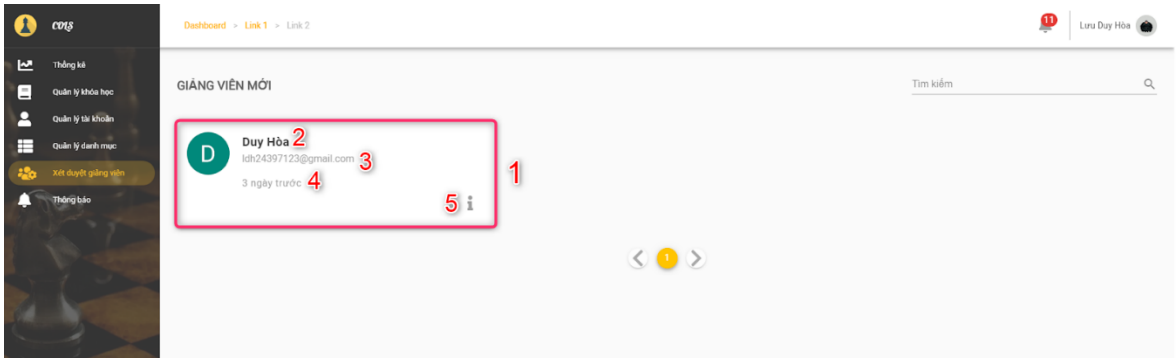


Figure 99: List of New Instructor

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	instructorItem	Instructor's item	Yes	Yes	Block	Card	N/A
2	instructorName	Instructor's name	Yes	Yes	Text	String	N/A
3	instructorEmail	Instructor's email	Yes	Yes	Text	String	N/A
4	createdTimeCountDown	Instructor's created time count down	Yes	Yes	Text	String	N/A

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
5	btnViewInfo	View instructor details	N/A	Show instructor details page

Table 101: List of New Instructor – Fields and Buttons/Hyperlinks

5.3.2.2 Instructor Details

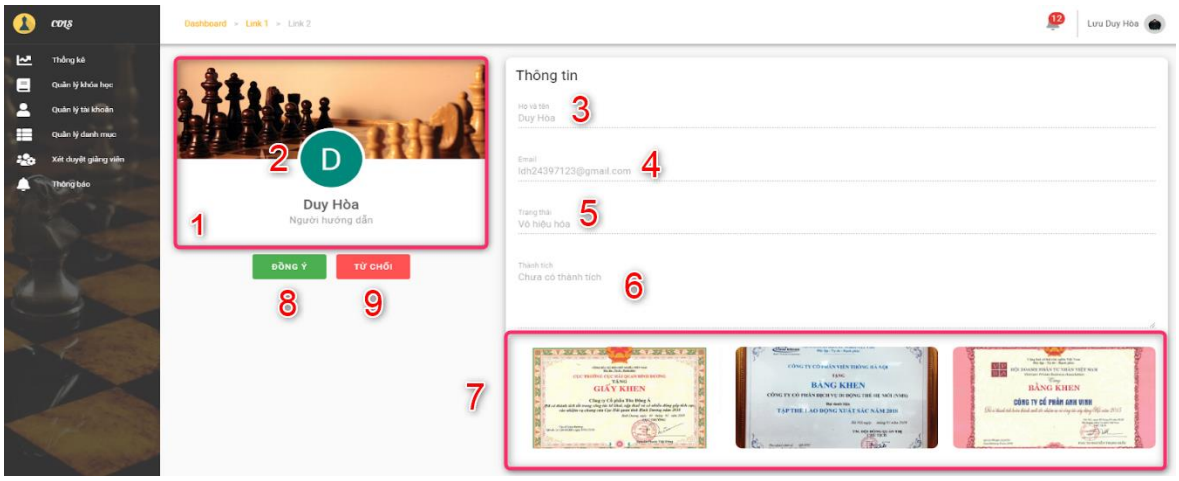


Figure 100: Instructor Details

FIELDS							
No	**Field Name**	**Description**	**Read only**	**Mandatory**	**Control Type**	**Data Type**	**Length**
1	summary	Instructor's summary information	Yes	Yes	Block	Card	N/A
2	instructorImage	Instructor's image	Yes	Yes	Image	File	N/A
3	instructorName	Instructor's name	Yes	Yes	Text	String	N/A
4	instructorEmail	Instructor's email	Yes	Yes	Text	String	N/A
5	instructorStatus	Instructor's status	Yes	Yes	Text	String	N/A
6	instructorAchievement	Instructor's achievement	Yes	Yes	Text	String	N/A
7	instructorCertificate	Instructor's certificates	Yes	Yes	Thumbnails	File	N/A

BUTTONS/HYPERLINKS				
No	**Function**	**Description**	**Validation**	**Outcome**
8	btnApprove	Approve instructor button	N/A	Approve new instructor
9	btnReject	Reject instructor button	N/A	Reject new instructor

Table 102: Instructor Details - Fields

6. Database Design

6.1 Entity relationship diagram (ERD)[3]

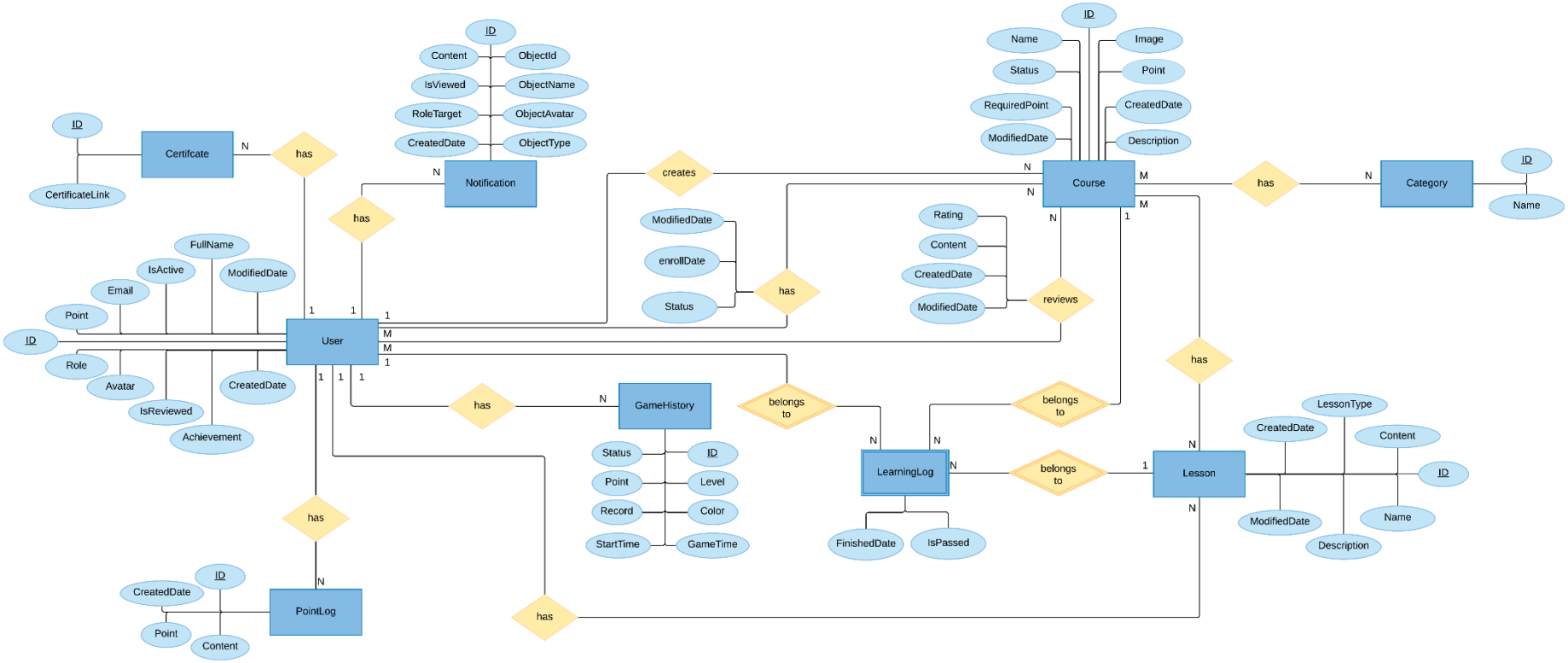


Figure 101: Entity relationship diagram (ERD)

6.2 Entity dictionary

Entity Data Dictionary: describe content of all entities	
Entity name	**Description**
User	Contains the user's information
Certificate	Contains certificates's link for Instructor
Notification	Contains User's notification
PointLog	Contains log about point transaction of Learner.
GameHistory	Contains the game history's information.
Course	Contains the course's information
Category	Contains the category's information
Lesson	Contains the lesson's information
LearningLog	Contains learning log for Learner

Table 103: Entity data dictionary

7. Algorithms

7.1 Convert chess move list from database to display list

7.1.1 Definition

In our interactive lesson or exercise, the content is a chess game with existing move list, we need to display all chess move for the learner, and learner can click on each move to view the analysis of that chess puzzle. Moreover, chess game maybe contains more than 1 move for certain chess puzzle. Our system approves that 1 puzzle, a learner can view 2 moves, game move and another move from analyst, and we limit this action in 2 times, it means, in another move, we only have 1 another move again or the move depth is 3.

7.1.2 Define problem

From above requirement, we used an array to display chess move for the learner, each object in array is a turn include: turn index, depth of move turn, if depth is 1, object will contain white move and black move and chess move information, such as: move direction, FEN string,… is store in every move, if the depth is 2, the object will contain an array of depth 2 move, and it is same with depth is 3. Then we use linked list to store chess move with JSON format in the database, and we need to convert move linked list to displayed list. For example:

6	♗g2	♗e7
7	O-O	O-O
7	...	b6
8	♖e1	♗b7
9	c3	
9	e5	♘d7
10	♘f1	♕c7
11	♗f4	h6
12	h4	O-O-O
9	...	♕c7

Figure 102: Target Move List


```
{
"id": 13,
"move": "O-O",
"content": "",
"moveDirection": "e1g1",
"fen": "r1bqk2r/pp2bppp/2n1pn2/2pp4/4P3/3P1NP1/PPPN1PBP/R1BQ1RK1 b kq - 4 7",
"preId": 12
},
{
"id": 14,
"move": "O-O",
"content": "",
"moveDirection": "e8g8",
"fen": "r1bq1rk1/pp2bppp/2n1pn2/2pp4/4P3/3P1NP1/PPPN1PBP/R1BQ1RK1 w - - 5 8",
"preId": 13
},
```


Figure 103: Sample move list from database

Move list dictionary	
Key	Description
id	Move Identifier
move	Chess Move in SAN, use for display move to the learner
content	Description for puzzle follow with a move
moveDirection	Move direction include source square, target square and maybe it contains promotion piece name. Chess programming needs to move directly to generate a new position.
fen	FEN String to present for current a chess position

preId	Previous move identifier

Table 104: Move list dictionary

7.1.3 Solution

To display chess move history in array list to the client. We need to take 3 steps:

- We will implement an array list to simple linked list by adding 1 attribute is the next step to each element.
- After we have a simple linked list, we will use Depth First Search (DFS) Algorithm.
- We will save this tree in array again but with our format for rendering

Firstly, we need to implement a simple linked list because we need to know what is the next steps of any step.

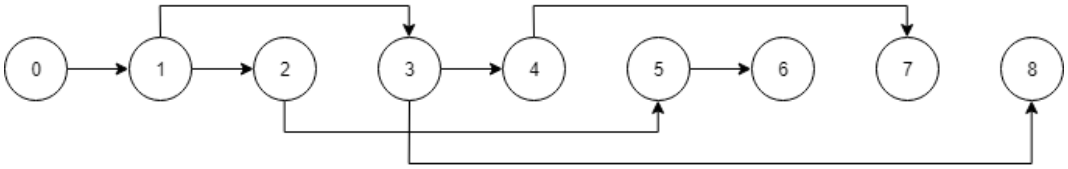


Figure 104: Example simple linked list

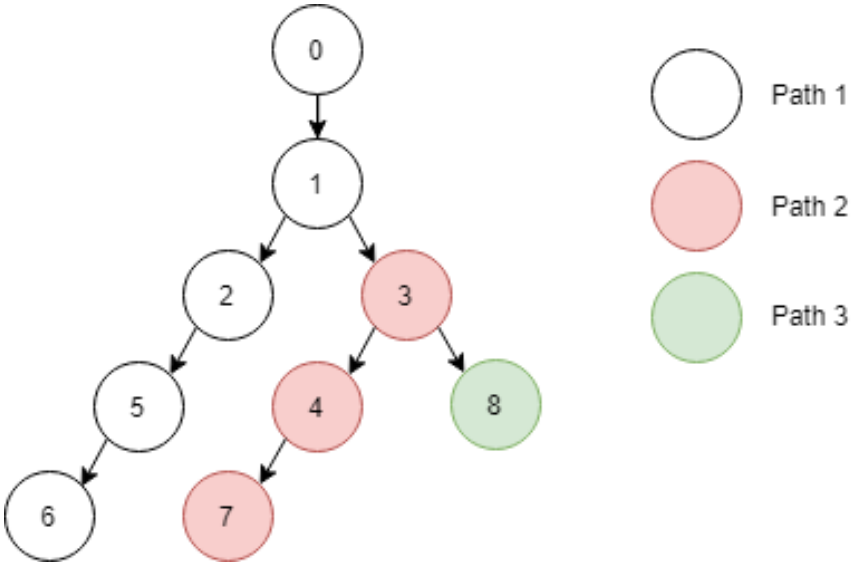


Figure 105: Example tree

With this tree, we have to find each path of the tree by using Recursion to implement Depth First Search (DFS) Algorithm. Of course, Breath First Search (BFS) is not suitable in this case.

Finally, while using DFS, we will save each node to element in the array with our format for rendering like this:

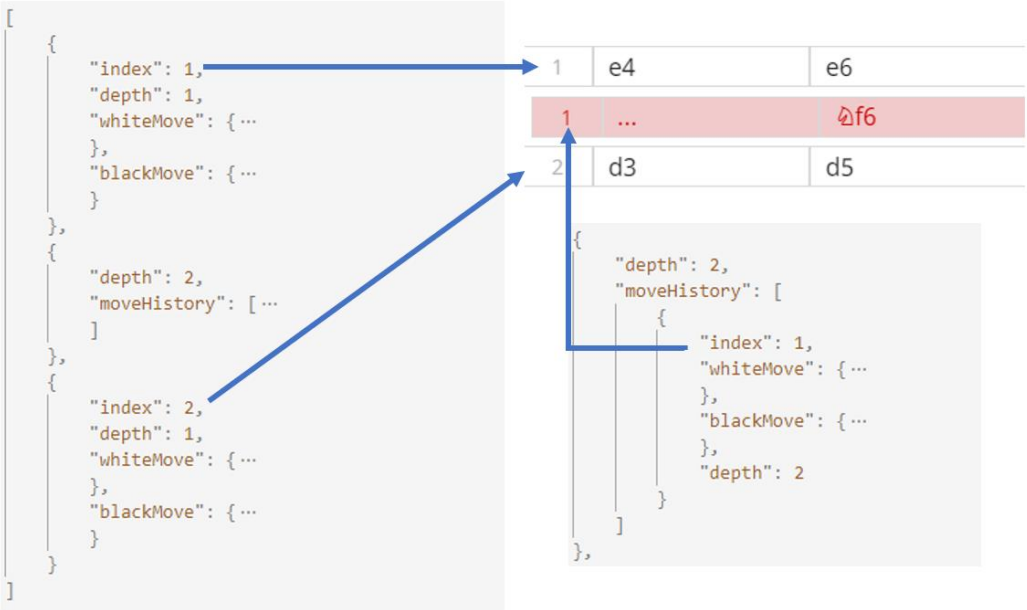


Figure 106: Move list result

7.1.4 Complexity

The complexity of this algorithm is: O(N)

7.2 Building chess bot

7.2.1 Definition

In COLS, we must build a chess bot, which can autoplay chess with the learner to improve their skills. We also use this bot to play with the learner in exercises.

7.2.2 Define problem

In a chess game, when a move is performed, there will have more possible game have existed. The number of game will be increased after more moves. Then we cannot use exhausted method to get the best move, it will take the memory very much.

When human play chess, they also calculate move with some of the material properties, such as:

- Controlling centre squares are better than controlling unoccupied squares.
- Double pawns are not good, they cannot defend each other.

Nowadays, there are more chess programming, they combine the chess material properties and computed methods to analyse chess move in computer. In COLS, we also use existing chess engine to get the best move for a specific chess position, which is called Stockfishjs.

7.2.3 Solution

In building play chess feature, we use stockfishjs chess engine to calculate the best move, Chessjs to handle the game and validates the player move. We also use Chessground library to show Chessboard for the learner.

There are some concepts in chess programming, then we will discuss an overview of them.

<u>Chessboard:</u> A chessboard will have 64 squares represents an 8x8 grid. In chess programming, they call file for each board column and rank for each board row. It

means a chessboard contains 8 ranks: 1-8 and 8 columns: a-h. We also use rank and file to record a square, such as a1, b3, e6,…, it is called Algebraic notation (AN).

8	a8	b8	c8	d8	e8	f8	g8	h8
7	a7	b7	c7	d7	e7	f7	g7	h7
6	a6	b6	c6	d6	e6	f6	g6	h6
5	a5	b5	c5	d5	e5	f5	g5	h5
4	a4	b4	c4	d4	e4	f4	g4	h4
3	a3	b3	c3	d3	e3	f3	g3	h3
2	a2	b2	c2	d2	e2	f2	g2	h2
1	a1	b1	c1	d1	e1	f1	g1	h1
	a	b	c	d	e	f	g	h

Figure 107: A chessboard with square in AN

<u>FEN String:</u> A FEN string is used to mark current position in a chess game, which will contain 6 parts – pieces position, move turn, castle tracking, en passant target, half move clock to handle 50 moves rule, and full move number.

- The first part in FEN String will use to represent chess pieces position by their name: **K**ing, **Q**ueen, **R**ook, **B**ishop, K**N**ight and **P**awn. The black pieces will be noted by lowercase (kqrbnp) and uppercase for white pieces. They will note chess pieces from rank 8 to rank 1 for each rank, and ranks will be joined at "/". If a square which contains pieces, it will be noted by pieces name and empty square will be noted by number. When a rank has adjacent empty squares, the number will be incremental until a piece square, for example:

Figure 108: Sample rank in FEN String

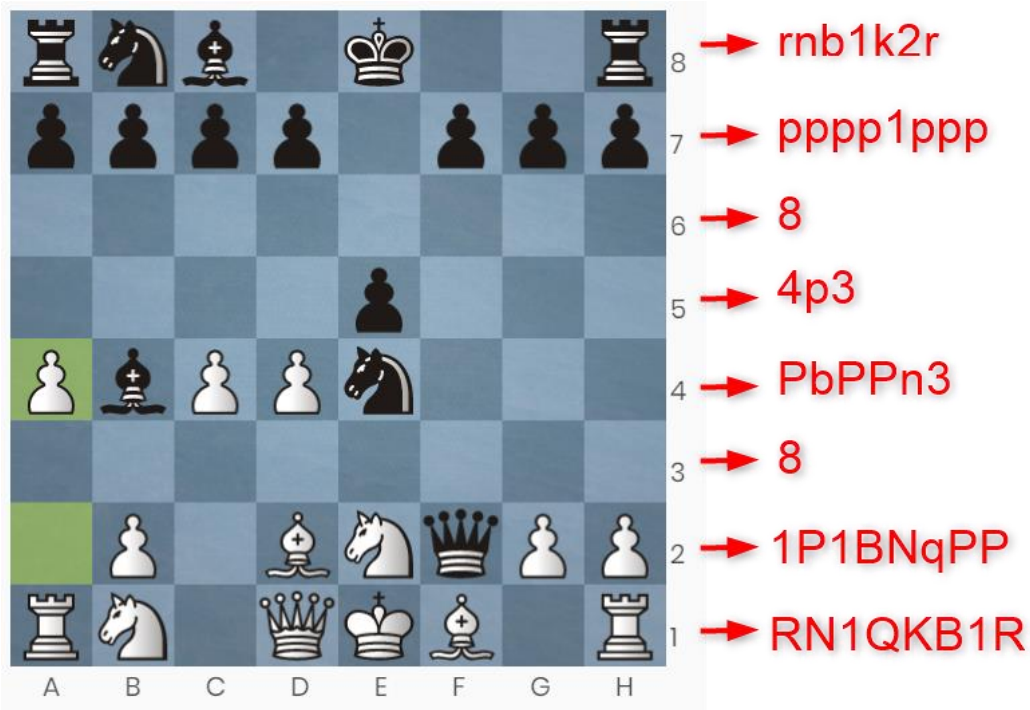


Figure 109: Sample Position in FEN String

(The above chess position will be noted:
rnb1k2r/pppp1ppp/8/4p3/PbPPn3/8/1P1BNqPP/RN1QKB1R)

- The second part in FEN String will represent for move turn: "w" for the white turn in next move and "b" for the black turn.
- The third part is castling tracking, each character is the wing which player can use castling move: K (White King), Q (White Queen), k (Black King), q (Black Queen).
- The fourth part will display an en passant target square if a player performs a two squares move or "-" for other moves. It means an en passant target move will be a move in rank 3 or 6.
- The fifth part is the total of half move since the last capture or pawn advance. It will be returned to 0 after a capture or pawn advance move and it is used to check 50 moves rule.
- The last part is the number of a full move in-game. It is count after a black's move.

FEN String is important data, which can represent all states in the current game. In our chess bot, we use all above chess libraries follow this flow:

- When the game is started, the all chess libraries: chessjs, chessground, stockfishjs will have initial FEN String:
  rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1
- For player turn:
  - When a move is performed in the chessboard, Chessground will generate a move, which contains source square and target square (and promotion pieces for promotion move in lowercase), for example e2e4 or f7f8q (f7 to f8, promote to Queen).
  - The move will be sent to chessjs and chessjs will generate new FEN String by it. Chessground will render a new position from new FEN String. User can see new game position.

- For bot turn, we use UCI command to communicate with Stockfishjs to get the best move, the UCI command:
  position startpos moves [string of moves from start game]
  If the string of moves is empty, the engine will calculate from start game, else it will calculate game from the game position in the last move of string. The result will be the best move with move direction (source square, target square and maybe promotion piece name). We also use this command, it will use FEN String for analyzing in custom position:
  position fen <fen string> moves [string of moves from start game]
- With the above UCI command, we need to add move after each half move.
- We also use chessjs to generate new FEN String and re-render new position.
- All the above step will repeat until the game is over. Chessjs lib will check FEN String to evaluate game is playing or over. After half move, chessjs also check possible moves and we will use possible moves to validate player move, if the move is not included, the player needs to perform again.

For building different bot level, we call different engine-level by this command:
setoption name Skill Level value <skill_level>

7.2.4 Flow chart

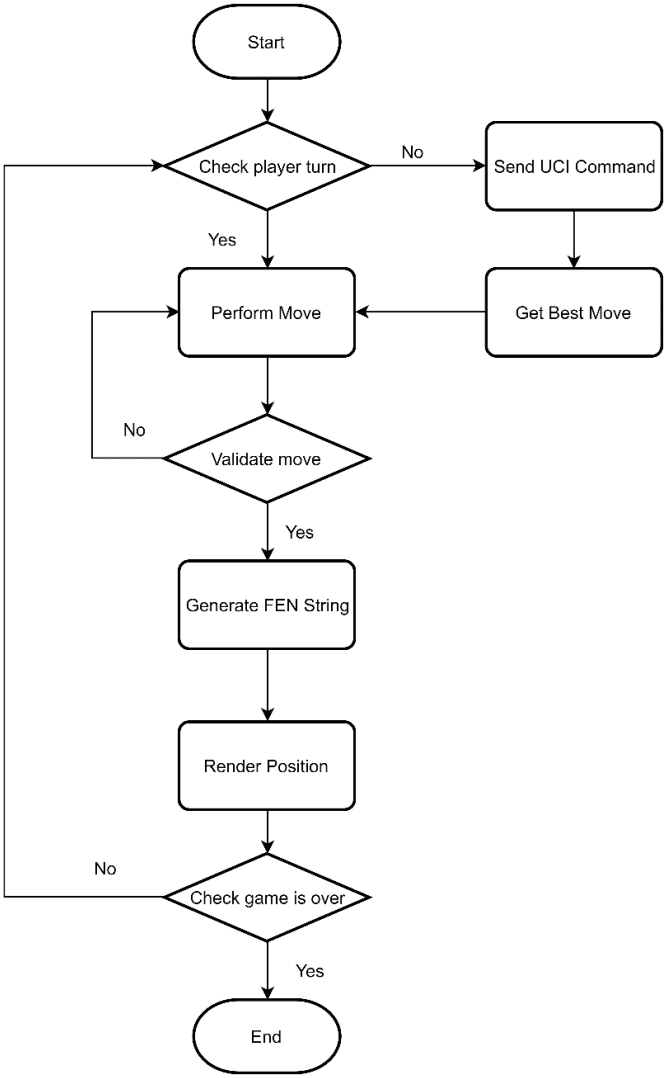


Figure 110: Building Chess Bot Flow chart

7.3 Building Elo rating system

7.3.1 Definition

In COLS, we must evaluate the learner's level by playing chess with the bot. Therefore, we need a standard rating system, which approved by FIDE (Fédération internationale des échecs).

7.3.2 Define problem

In COLS, the learner will start with specially point based on their chosen level. Whenever learner complete playing chess with the bot, the system will update point of learner depend on a final result chess game.

Each level of stockfishjs chess bot will have a specially Elo point and learner point will be calculated following the Elo rating system.

7.3.3 Solution

In the learner's rankings, we use the International Chess Association (FIDE) rankings[15] as the standard and readjust it to match the ranking of the system. Each rank will correspond to an Elo range:

- Beginner (0 ~ 1000 Elo points)
- Minor (1000 ~ 1200 Elo points)
- Intermediate (1200 ~ 1400 Elo points)
- Major (1400 ~ 1600 Elo points)
- Master(> 1600 Elo points)

When users create an account will allow them to choose their current level. Then the system will fill in the corresponding points for them.

In classifying each level of the bot in the Elo system. We use the CCRL 4040 calculation to help evaluate the bot's Elo level when learners play:

Rank	Name	Elo	+	-	games	score	oppo.	draws
1	Stockfish 10 x64 1T20	3495	102	102	400	99%	2804	2%
2	Stockfish 10 x64 1T19	2855	32	32	400	45%	2965	26%
3	Stockfish 10 x64 1T18	2825	32	32	400	41%	2972	27%
4	Stockfish 10 x64 1T17	2793	32	32	400	36%	2980	27%
5	Stockfish 10 x64 1T16	2745	23	23	800	50%	2788	24%
6	Stockfish 10 x64 1T15	2677	30	30	400	60%	2602	26%
7	Stockfish 10 x64 1T14	2652	30	30	400	56%	2608	26%
8	Stockfish 10 x64 1T13	2552	31	31	400	39%	2633	21%
9	Stockfish 10 x64 1T12	2458	24	24	800	50%	2444	14%
10	Stockfish 10 x64 1T11	2370	33	33	400	65%	2253	14%
11	Stockfish 10 x64 1T10	2269	32	32	400	49%	2279	12%
12	Stockfish 10 x64 1T09	2181	33	33	400	36%	2301	13%
13	Stockfish 10 x64 1T08	2105	25	25	800	49%	2108	8%
14	Stockfish 10 x64 1T07	2050	35	35	400	66%	1910	7%
15	Stockfish 10 x64 1T06	1950	34	34	400	51%	1935	5%
16	Stockfish 10 x64 1T05	1862	34	34	400	39%	1957	4%
17	Stockfish 10 x64 1T04	1722	29	29	800	52%	1683	3%
18	Stockfish 10 x64 1T03	1589	37	37	400	68%	1408	2%
19	Stockfish 10 x64 1T02	1427	36	36	400	48%	1449	2%
20	Stockfish 10 x64 1T01	1302	38	38	400	32%	1480	1%
21	Stockfish 10 x64 1T00	1181	42	42	400	19%	1510	0%

The realistic part of the ellipse after the learner completes the game of chess with the bot. It includes the probability of making ellipse commands between learners and bots and finding K-factors.

Probability of ellipse between learner and bot In which learners are A and bots are B:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}}$$

$$E_B = \frac{1}{1 + 10^{(R_A - R_B)/400}}$$

E_A: expected score of A

E_B: expected score of B

R_A: rating of A (current elo point)

R_B: rating of B (current elo point)

Note that:

$$\boldsymbol{E_A + E_B = 1}$$

$$R'_A = R_A + K(S_A - E_A)$$

R'_A: new rating of A

S_A: actually scored (1 for win , 0 for lose and drawn for 0.5)

K: k-factors

FIDE used the following ranges for K[15]:

- K = 40 for a player new to the rating list until he has completed events with at least 30 games.
- K = 20 as long as a player's rating remains under 2400.
- K = 10 once a player's published rating has reached 2400 and remains at that level subsequently, even if the rating drops below 2400.

7.4 Finding Similar User

7.4.1 Definition

In order to make a recommendation for learners based on existing data, we use Collaborative filtering technique by memory-based. Firstly, we need to find similar users with the current user.

7.4.2 Define problem

The existing data is a rating of learners for their courses. We would calculate similarity based on the interest of learners with courses.

7.4.3 Solution

We apply cosin similarity to calculate similar between 2 users:

$$similarity(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2}\sqrt{\sum_{i=1}^{n} {B_i}^2}}$$

A_i: Rating in course i of user A

B_i: Rating in course i of user B

Result range: [0,1] with 1 is the most similar

7.4.4 Example

There are ratings of 4 users for course 1 to 5:

Course	User A	User B	User C	User D
1	4.0	5.0	4.5	5.0
2	5.0	null	4.0	4.5

3	4.0	4.0	null	5.0
4	3.0	3.5	4.0	null
5	null	3.0	3.0	4.5

Similarity between User A and other User:

$$similarity(A,B) = \frac{4*5+4*4+3*3.5}{\sqrt{4^2+5^2+4^2+3^2}\sqrt{5^2+4^2+3.5^2+3^2}} = 0.78006$$

$$similarity(A,C) = \frac{4*4.5+5*4+3*4}{\sqrt{4^2+5^2+4^2+3^2}\sqrt{4.5^2+4^2+4^2+3^2}} = 0.7864$$

$$similarity(A,D) = \frac{4*5+5*4.5+4*5}{\sqrt{4^2+5^2+4^2+3^2}\sqrt{5^2+4.5^2+5^2+4.5^2}} = 0.80869$$

7.4.5 Complexity

The Complexity is: O(n)

7.5 Course Rating Prediction (User-based filtering)

7.5.1 Definition

From the above result, we perform next step is predicting ratings for courses which user has not been learned based on similar users.

7.5.2 Define problem

Each user has different courses, but if they have large similarity, we can predict the rating for courses which they have not been learned.

7.5.3 Solution

Calculating the weighted average based on rating and similarity of top n user:

$$R = \frac{(\sum_{u=1}^{n} R_u S_u)}{\sum_{u=1}^{n} S_u}$$

R: Predicted rating of course

R_u: Rating in course of user u

S_u: Similarity between current user with user u

7.5.4 Example

Course	User A	User B	User C	User D
1	4.0	5.0	4.5	5.0
2	5.0	null	4.0	4.5
3	4.0	4.0	null	5.0
4	3.0	3.5	4.0	null
5	null	3.0	3.0	4.5

We will use above data and assump that User B, C, D is top 3 similar user of A. The similar between A and them are sim(A,B) = 0.78, sim(A,C) = 0.7864 and sim(A,D) = 0.809.

The predicted rating of A for course 5 is:

$$R = \frac{3*0.78+3*0.7864+4.5*0.809}{0.78+0.7864+0.809} = 3.51$$

7.5.5 Complexity

The Complexity is: O(n)

7.6 Item-based filtering

7.6.1 Definition

In COLS, we also calculate similar between system courses and user enrolled courses.

7.6.2 Define problem

When the user views a specific course, the system will find similar courses with current course to suggest for the user. The suggested course must be the same with learner skill level.

7.6.3 Solution

We also using cosin similarity with special parameter to calculate similar course. Courses in the system will have similar attributes:

Attribute	Course A (current course)	Course B
Same instructor	1	x
Same category	1	y
Course rating	r_A	r_B

x: value is 0 if course B has different instructor and 1 if course B has same instructor

y = $\frac{number\ of\ same\ categories\ between\ course\ A\ and\ course\ B}{number\ of\ categories\ in\ course\ A}$

r_A: Average rating of course A

r_B: Average rating of course B

$$similarity(A,B) = \frac{x + y + r_A * r_B}{\sqrt{{r_A}^2 + 2}\sqrt{x^2 + y^2 + {r_B}^2}}$$

Result range: [0,1] with 1 is the most similar

7.6.4 Example

Attribute	Course A (current course)	Course B	Course C
Same instructor	1	0	1
Same category	1	0.5	0.33
Course rating	4	4.5	4

Similarity between Course A and other courses:

$$similarity(A,B) = \frac{0.5 + 4.5 * 4}{\sqrt{4^2 + 2}\sqrt{0.5^2 + 4.5^2}} = 0.46$$

$$similarity(A,C) = \frac{1 + 0.33 + 4 * 4}{\sqrt{4^2 + 2}\sqrt{1 + 0.33^2 + 4^2}} = 0.98$$

Course C is almost identical to A than the other course because the similarity is larger.

7.6.5 Complexity

The Complexity is: O(n)

E. System Implementation & Test

1. Introduction

1.1 Overview

This section describes the approach and methodologies for the testing phase of COLS. It provides in the detail necessary information about the

implementation and testing procedure of the system included test plans, test cases, test result, test environments and pass/fail criteria as a checklist to cover possible cases.

1.2 Test approach

- **Goal:** Test all features in COLS based on the core flow.
- **Method:** black-box testing
- **Technique:** check list

2. Database Relationship Diagram

2.1 Physical diagram

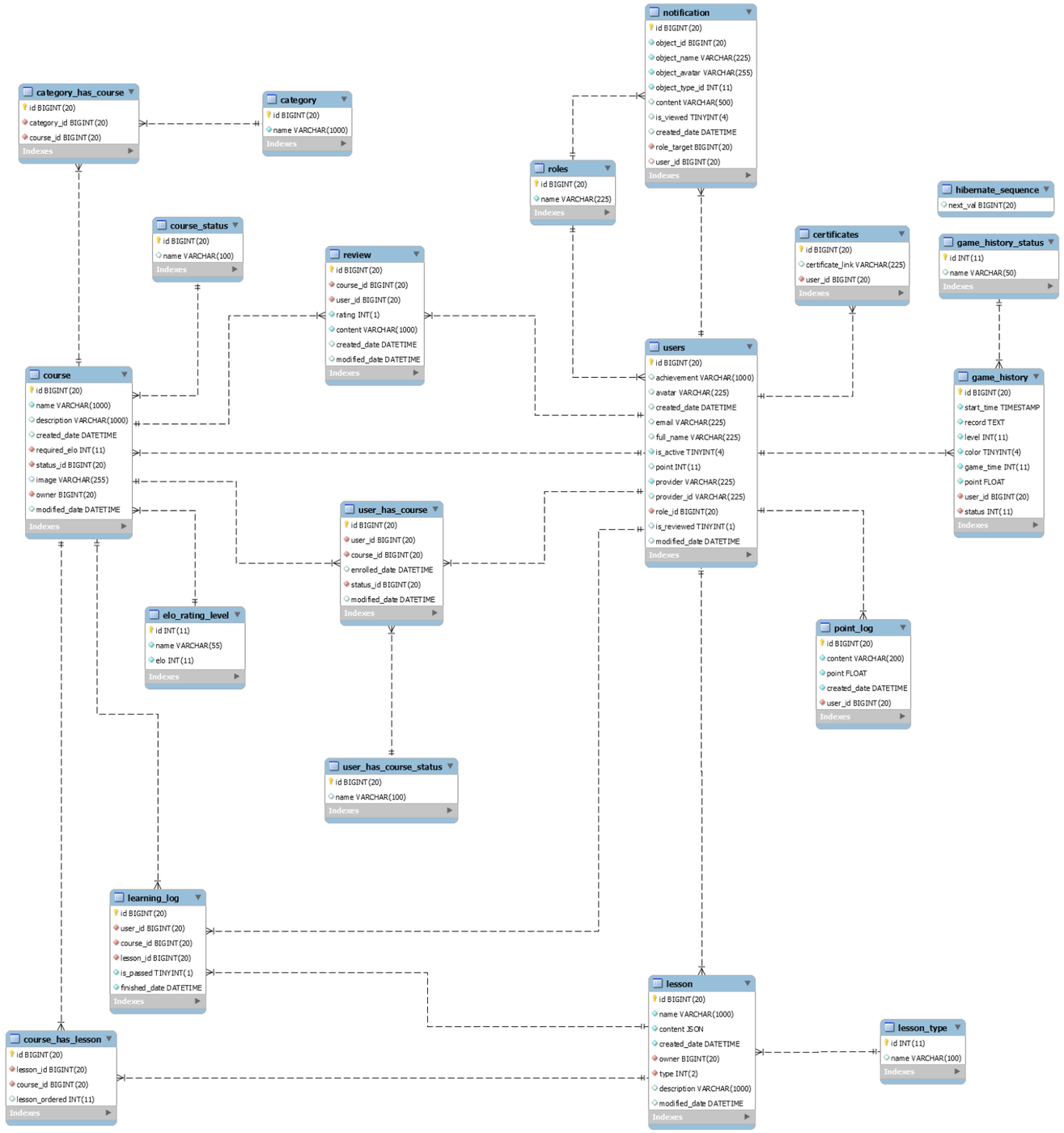


Figure 111: Physical diagram

2.2 Data dictionary

DATA DICTIONARY: DESCRIBE CONTENT OF ALL TABLES	
Table name	**Description**
users	Contains the user information
roles	Contains the user role name
point_log	Contains the log of user point transactions
notification	Contains the notification for each user
certificates	Contains the certificates for instructor
game_history	Contains the game history for learner
game_history_status	Contains the status name of game
review	Contains the course review information
course	Contains the course information
course_status	Contains the status name for a course
elo_rating_level	Contains the level name based on elo rating
category	Contains the category information
category_has_course	Contains the relationship between category and course
user_has_course	Contains the relationship between course and user
user_has_course_status	Contains the status name of course for user
lesson	Contains the lesson information
lesson_type	Contains the lesson type name
course_has_lesson	Contains the relationship between course and lesson
learning_log	Contains the learning progress information of a learner
hibernate_sequence	Table of Hibernate for tracking auto-increment ID

Table 105: Data Table Dictionary

Table Name	Attributes	Description	Domain	Null
user	id	Unique identifier of a user	BIGINT(20)	NO
	achievement	Instructor's achievement	VARCHAR(1000)	YES
	avatar	User's avatar URL	VARCHAR(225)	YES
	created_date	Created time of a user	DATETIME	YES
	email	User's email	VARCHAR(225)	YES
	full_name	User's full name	VARCHAR(225)	YES
	is_active	Current user's status	TINYINT(4)	NO
	point	Learner's point	FLOAT	YES
	provider	User's social network provider name	VARCHAR(225)	NO
	provider_id	User's social network provider identifier	VARCHAR(225)	YES
	role_id	Identifier of the user's role	BIGINT(20)	NO

	is_reviewed	Tracking instructor is reviewed or not	TINYINT(1)	NO
	modified_date	User information updated time	DATETIME	YES
roles	id	Unique identifier of a role	BIGINT(20)	NO
	name	Name of role	VARCHAR(225)	NO
point_log	id	Unique identifier of a point's log	BIGINT(20)	NO
	content	Point transaction's reason	VARCHAR(200)	NO
	point	Number of points	FLOAT	NO
	created_date	Log's time	DATETIME	NO
	user_id	Point owner's identifier	BIGINT(20)	NO
notification	id	Unique identifier of the notification	BIGINT(20)	NO
	object_id	Source object identifier	BIGINT(20)	NO
	object_name	Source object name	VARCHAR(225)	NO
	object_avatar	Source object avatar	VARCHAR(225)	NO
	object_type_id	Source object type identifier	INT(11)	NO
	content	Notification content	VARCHAR(500)	YES
	is_viewed	Tracking notification is viewed or not	TINYINT(4)	YES
	created_date	Notification time	DATETIME	YES
	role_target	Target role identifier	BIGINT(20)	NO
	user_id	Target user identifier	BIGINT(20)	NO
certificates	id	Unique indentifier of a certificate	BIGINT(20)	NO
	certificate_link	Instructor certificate image URL	VARCHAR(225)	YES
	user_id	Instructor's identifier	BIGINT(20)	NO
game_history	id	Unique identifier of a game history	BIGINT(20)	NO
	start_time	Game's start time	TIMESTAMP	NO
	record	Game's PGN	TEXT	NO
	color	Learner's colour	TINYINT(4)	NO
	game_time	Playing time for each player	INT(11)	NO
	point	Bonus point when ending game	FLOAT	NO
	user_id	Learner's identifier	BIGINT(20)	NO
	status	Game's status	INT(11)	NO
game_history_status	id	Unique identifier of a game status	INT(11)	NO

	name	Game status name	VARCHAR(50)	YES
review	id	Unique identifier of a review	BIGINT(20)	NO
	course_id	Course's identifier which contains the review	BIGINT(20)	NO
	user_id	Reviewer's identifier	BIGINT(20)	NO
	rating	Review's rating point	INT(1)	NO
	content	Review's content	VARCHAR(1000)	NO
	created_date	Review's time	DATETIME	YES
	modified_date	Review's updated time	DATETIME	YES
course	id	Unique identifier of a course	BIGINT(20)	NO
	name	Course's name	VARCHAR(1000)	NO
	description	Course's description	VARCHAR(1000)	YES
	created_date	Course's created time	DATETIME	YES
	required_elo	Course's required level identifier	INT(11)	NO
	status_id	Current course's status identifier	BIGINT(20)	NO
	image	Course's cover image URL	VARCHAR(255)	YES
	owner	Course's owner identifier	BIGINT(20)	NO
	modified_date	Course's updated time	DATETIME	YES
course_status	id	Unique identifier of a course's status	BIGINT(20)	NO
	name	Course's status name	VARCHAR(100)	YES
category	id	Unique identifier of a category	BIGINT(20)	NO
	name	Category's name	VARCHAR(1000)	NO
category_has_course	id	Unique identifier of a relationship between course and category	BIGINT(20)	NO
	category_id	Category identifier	BIGINT(20)	NO
	course_id	Course identifier	BIGINT(20)	NO
elo_rating_level	id	Unique identifier of a level	INT(11)	NO
	name	Level's name	VARCHAR(55)	NO
	elo	Level's required Elo	INT(11)	NO
user_has_course	id	Unique identifier of a relationship between user and course	BIGINT(20)	NO
	user_id	User identifier	BIGINT(20)	NO
	course_id	Course identifier	BIGINT(20)	NO

	enrolled_date	Enrol's time	DATETIME	YES
	status_id	Status identifier between course and user	BIGINT(20)	NO
	modified_date	Relationship's status updated time	DATETIME	YES
user_has_ course_status	id	Unique identifier of a status name for the relationship between user and course	BIGINT(20)	NO
	name	Status's name	VARCHAR(100)	YES
lesson	id	Unique identifier of a lesson	BIGINT(20)	NO
	name	Lesson's name	VARCHAR(1000)	NO
	content	Lesson's content	JSON	NO
	created_date	Lesson's created time	DATETIME	NO
	owner	Lesson's owner identifier	BIGINT(20)	NO
	type	Lesson's type identifier	INT(2)	NO
	description	Lesson's description	VARCHAR(1000)	YES
	modified_date	Lesson's updated time	DATETIME	YES
lesson_type	id	Unique identifier of a lesson type	INT(11)	NO
	name	Lesson type's name	VARCHAR(100)	YES
course_has_ lesson	id	Unique identifier of a relationship between course and lesson	BIGINT(20)	NO
	lesson_id	Lesson identifier	BIGINT(20)	NO
	course_id	Course identifier	BIGINT(20)	NO
	lesson_ordered	Lesson ordered in course	INT(11)	YES
learning_log	id	Unique identifier of a learning log	BIGINT(20)	NO
	user_id	Learner identifier	BIGINT(20)	NO
	course_id	Course identifier	BIGINT(20)	NO
	lesson_id	Lesson identifier	BIGINT(20)	NO
	is_passed	Tracking lesson is passed by the learner	TINYINT(1)	NO
	finished_date	Lesson finished time	DATETIME	NO
hibernate_ sequence	next_val	Next value of auto-increment key	BIGINT(20)	YES

Table 106: Data Table Description

3. Test plan

The purpose of testing is to ensure COLS reaches technical requirements and business rules. This section describes the core flow for each actor in COLS and test cases for each feature inflows. The test cases also are updated as required for each release.

3.1 Features to be tested

- Guest: Sign up, Sign in
- Learner: View published courses, Search courses, View course details, Enrol course, View enrolled course, Learn course, Create review, View list of suggested course, Play chess with the bot
- Instructor: Create course, Create interactive lesson, Create uninteractive lesson, Create exercise, Edit course status (Submit course)
- Admin: Create category, Review instructor, Review new course

3.2 Features not to be tested

- Guest: Filter courses
- Authorized User: View profile, Edit profile, Sign out
- Learner: Edit review, Remove review, View learning progress
- Instructor: Edit course, Edit interactive lesson, Edit Uninteractive lesson, Edit exercise, Remove lesson (exercise), View course reports
- Admin: Edit category, Remove category, View course list, Unpublish course, View user list, Active/Deactivate user, View reports

4. System Testing Test Case

4.1 Core Flow

4.1.1 Guest Core Flow

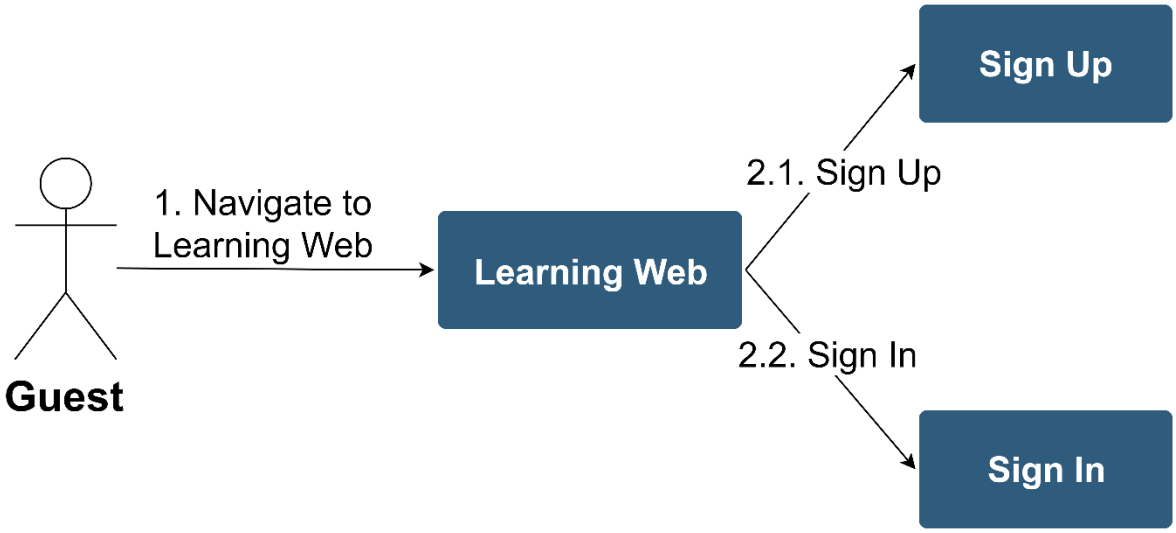


Figure 112: <Guest> Core Flow

4.1.2 Learner Core Flow

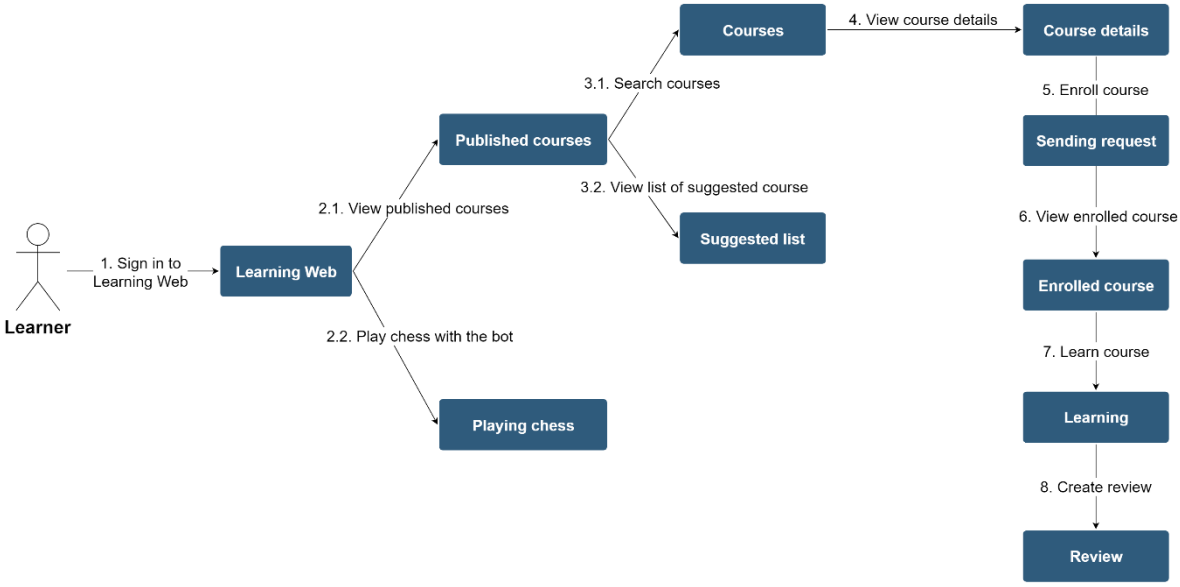


Figure 113: <Learner> Core Flow

4.1.3 Instructor Core Flow

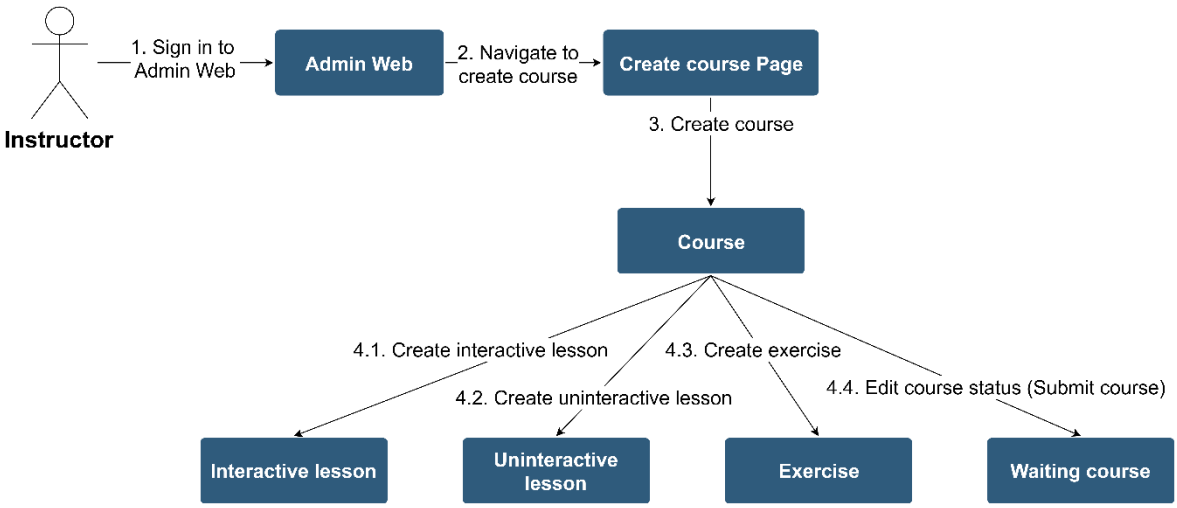


Figure 114: <Instructor> Core Flow

4.1.4 Admin Core Flow

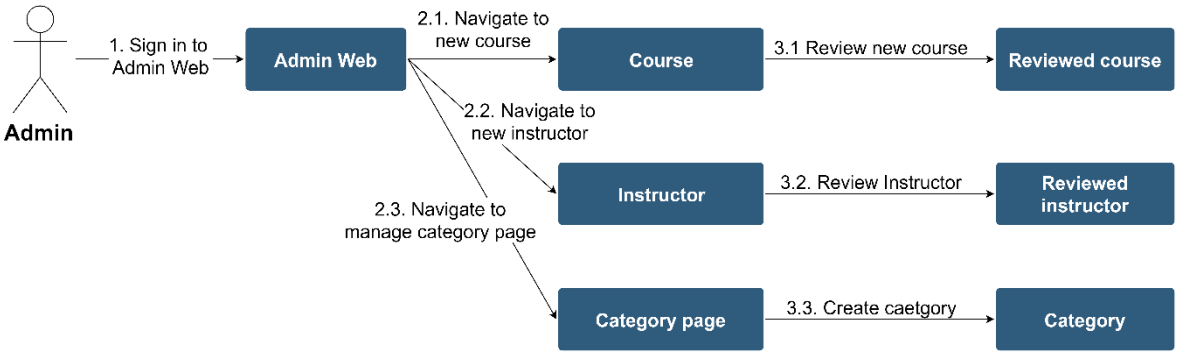


Figure 115: <Admin> Core Flow

4.2 Test case

4.1.1 <Guest> Sign Up – Sign In

ID	Test Case Description	Test case procedure	Expected output	Result	Test Date
SIGN UP					
SU_1	Guest inputs empty name	1. Guest signs in to system in the first time, system redirects to register form. 2. Guest inputs full name: “”.	System shows an error message: “Họ và tên không được bỏ trống”.	Pass	20/09/2019
SU_2	Guest inputs name more than 225 characters	1. Guest signs in to system by new email, system redirects to register form. 2. Guest inputs full name: “*jasdhasdhjakshdkjdhaskdhaskuiowquiowqflkdsjfkdshfdskfhkjsdhfsjdkhfkjdshfkjhdskjfhkjsdhfkjhdjskhfkjdshkkfhsdjkfhksdfsdfsdffsdfsdffsdfsdfdsdfsdewrwerewrwerdglaksdlaposdwqesfjkjsldjaskldhwbrbkjxchaksdsdfsdfsafafdafsadfsdfsdfsdfsdfaewqw*”.	System shows error message: “Họ và tên không được nhiều hơn 225 kí tự”.	Pass	20/09/2019
SU_3	Guest inputs name less than 6 characters	1. Guest signs in to system by new email, system redirects to register form. 2. Guest inputs full name: “Na”.	System shows an error message: “Họ và tên không được ít hơn 6 kí tự”.	Pass	20/09/2019
SU_4	Guest upload wrong image format	1. Guest signs in to system by new email, system redirects to register form. 2. Guest upload a file: test.txt.	System shows an error message: “Hình ảnh không đúng định dạng”	Pass	20/09/2019

SU_5	Guest signs up without agreement with system clauses.	1. Guest signs in to system by new email, system redirects to register form. 2. Guest does not check agreement with system clause.	System shows an error message: "Bạn phải đồng ý để tiếp tục"	Pass	20/09/2019
SU_6	Guest signs up successful with learner role	1. Guest signs in to system by new email, system redirects to register form. 2. Guest inputs valid learner information. 3. Guest submit signed-up form.	System redirects to Home page and shows snack bar with message: "Chúc mừng bạn đã đăng ký trở thành học viên thành công.".	Pass	20/09/2019
SU_6	Guest signs up successful with instructor role	1. Guest signs in to system by new email, system redirects to register form. 2. Guest inputs valid instructor information. 3. Guest submit signed-up form.	System redirects to Home page and shows snack bar with message: "Chúc mừng bạn đã đăng ký trở thành người hướng dẫn thành công. Vui lòng chờ hệ thống xét duyệt tài khoản của bạn.".	Pass	20/09/2019
SIGN IN					
SI_1	Guest signs in to Learning website by instructor account.	Guest signs in to Learning website by this account: "instructor.cols@gmail.com".	System returns to current page without changing.	Pass	20/09/2019
SI_2	Guest signs in to Learning website by admin account.	Guest signs in to Learning website by this account: "colsabcxyz@gmail.com".	System returns to current page without changing.	Pass	20/09/2019

SI_3	Guest signs in to Admin website by learner account.	Guest signs in to Admin website by this account: "cols.learner@gmail.com".	System returns to current page without changing.	Pass	20/09/2019
SI_4	Guest signs in to Learning website by learner account.	Guest signs in to Learning website by this account: "cols.learner@gmail.com".	System returns to current page with signed in user.	Pass	20/09/2019
SI_5	Guest signs in to Admin website by instructor account.	Guest signs in to Admin website by this account: "instructor.cols@gmail.com".	System redirects to dashboard page for instructor.	Pass	20/09/2019
SI_6	Guest signs in to Admin website by admin account.	Guest signs in to Admin website by this account: "colsabcxyz@gmail.com"	System redirects to dashboard page for admin.	Pass	20/09/2019

Table 107: Test cases <Guest>

4.1.2 <Learner> View Published Courses – Search Courses – View Course Details – Enrol Course – View Enrolled Course – Learn Course – Create Review – View List Of Suggested Course – Play Chess With The Bot

ID	Test Case Description	Test case procedure	Expected output	Result	Test Date
VIEW PUBLISHED COURSES					
VP_1	Learner view courses which were published in the system.	Learner opens the learning page from Home page.	System shows list of published course with basic information for each course: name, required skill level, description, instructor name, categories, rating and course image.	Pass	20/09/2019
SEARCH COURSES					

SC_1	Learner searches course with empty name	1. Learner input “” in search course field. 2. Learner clicks on “Tìm kiếm” button.	System shows an error message “Tên khóa học không được để trống”.	Pass	20/09/2019
SC_2	Learner searches courses by course’s name	1. Learner input “căn bản” in search course field. 2. Learner clicks on “Tìm kiếm” button.	System shows list of courses which name contains “căn bản”.	Pass	20/09/2019
VIEW COURSE DETAILS					
VD_1	Learner views a specific course	Learner clicks on “Xem chi tiết” button of any course.	System redirects to course page with course details: instructor information, number of enrolled learner, number of the lesson, list of the lesson, course description, required skill level, required point, Overview box and the lastest comment box, list of suggested courses and enrolled button.	Pass	20/09/2019
ENROL COURSE					
EC_1	Enrol course when not enough level	1. On the learning page, learner selects “Khóa học nâng cao” course. 2. Learner clicks on “Đăng ký khóa học” button.	System shows an error messages: “Bạn chưa đủ điều kiện để đăng kí khóa học này. Xin thử lại các khóa học khác!”	Pass	20/09/2019
EC_2	Enrol course with enough level	1. On the learning page, learner selects “Khóa học cờ vua căn bản” course. 2. Learner clicks on “Đăng ký khóa học” button.	System shows a message: “Đăng kí khóa học thành công and unlock lesson list. The “Đăng ký khóa học” button is changed to “Bắt đầu học” button.	Pass	20/09/2019

VIEW ENROLLED COURSE					
VE_1	Learner views an enrolled course	On the learning page, learner selects "Khóa học cờ vua căn bản" course.	System shows course details page and enables review area. Lessons are unlocked.	Pass	20/09/2019
LEARN COURSE					
LC_1	Learner learns course by "Bắt đầu học" button	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course. 2. Learner clicks on "Bắt đầu học" button.	System redirects to the next lesson which learner did not learn.	Pass	20/09/2019
LC_2	Learner learns by a specific lesson	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course. 2. Learner clicks on the "view Bài học 1" button.	System redirects to learning page with "Bài học 1" 's content	Pass	20/09/2019
LC_3	Learner backs to course details page	On the "Bài học 1" page, learner clicks on the exit button.	System redirects learner to course details page.	Pass	20/09/2019
LC_4	Learner navigates to previous lesson.	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course. 2. Learner clicks on the "view Bài học 2" button. 3. Learner clicks on "Bài trước" button.	System shows "Bài học 1" 's content.	Pass	20/09/2019
LC_5	Learner read interactive lesson by arrow buttons.	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course. 2. Learner clicks on the "view Bài học 1" button. 3. Learner clicks on the next arrow buton.	System shows next selected move with coresponsding chess position and maybe with move content.	Pass	20/09/2019

LC_6	Learner read interactive lesson by selects move	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course. 2. Learner clicks on the "view Bài học 1" button. 3. Learner clicks on any chess move.	System shows selected move with coresponsding chess position and maybe with move content.	Pass	20/09/2019
LC_7	Learner does wrong manual exercise	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course. 2. Learner clicks on the "view Bài tập 1" button. 3. Learner performs wrong moves.	System returns to previous move and maybe show wrong response.	Pass	20/09/2019
LC_8	Learner does right manual exercise	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course. 2. Learner clicks on the "view Bài tập 1" button. 3. Learner performs rights moves.	System performs next move and record that learners passed these exercise if they can performs all rights move.	Pass	20/09/2019
LC_9	Bot Exercise is ended without checkmate	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course. 2. Learner clicks on the "view Bài tập 2" button. 3. Learner performs rights moves. 4. Exercise is ended without checkmate.	System records that learners did not passed these exercise.	Pass	20/09/2019
LC_10	Bot Exercise is ended with checkmate	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course. 2. Learner clicks on the "view Bài tập 2" button. 3. Learner performs rights moves. 4. Learner checkmates.	System records that learners passed these exercise.	Pass	20/09/2019
LC_11	Learner request next interactive or	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course.	System redirects to "Bài tập 1" lesson, which is the next lesson	Pass	20/09/2019

	uninteractive lesson.	2. Learner clicks on the "view Bài học 2" button. 3. Learner clicks on the next lesson button.	of "Bài học 2" and marks "Bài học 2" lesson is passed.		
CREATE REVIEW					
CR_1	Learner creates review with empty rating	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course. 2. Learner clicks on the "view Bài học 2" button. 3. Learner inputs on the review text area: "Khóa học này cũng bình thường" and does not select rating points.	System shows an error message "Điểm đánh giá phải lớn hơn 0"	Pass	20/09/2019
CR_2	Learner creates valid review	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course. 2. Learner clicks on the "view Bài học 2" button. 3. Learner inputs on the review text area: "Khóa học này cũng bình thường" and selects rating points is 3.	Course review is re-calculated and review is shown on the top of list.	Pass	20/09/2019
CR_3	Learner creates second review	1. On the learning page, learner selects "Khóa học cờ vua căn bản" course, which learner reviewed. 2. Learner clicks on the "view Bài học 2" button. 3. Learner inputs on the review text area: "Khóa học này cũng bình thường" and selects rating points is 5.	System shows an error message "Đánh giá không thành công. Bạn đã đánh giá khóa học này rồi".	Pass	20/09/2019
VIEW LIST OF SUGGESTED COURSE					

VS_1	Learner view suggested list from learning page.	1. On the learning page, learner selects "Khóa học gợi ý" link.	System shows list of suggested courses for learner.	Pass	20/09/2019
VS_2	Learner view suggested list from a specific course.	1. On the learning page, learner selects "Khóa học cờ vua căn bản".	System shows list of suggested course which is similar with "Khóa học cờ vua căn bản" course.	Pass	20/09/2019
PLAY CHESS WITH THE BOT					
PC_1	Play chess with bot	1. Learner clicks on "Chơi thử" button. 2. Learner clicks on the start game button. 3. Learner inputs valid game information. 4. Learner clicks "Bắt đầu" button.	System close dialog and start a new game.	Pass	20/09/2019
PC_02	The chess game is completed	1. Learner was playing chess with bot	System shows dialog result and includes information about bonus reward points.	Pass	20/09/2019
PC_03	Countdown time of learner is over.	1. Learner was playing chess with bot	System shows dialog result about learner lose the game and includes information about bonus reward points.	Pass	20/09/2019
PC_04	Countdown time of bot is over.	1. Learner was playing chess with bot	System shows dialog result about learner win the game and includes information about bonus reward points.	Pass	20/09/2019

Table 108: Test cases <Learner>

4.1.3 <Instructor> Create Course – Create Interactive Lesson – Create Uninteractive Lesson – Edit Course Status

ID	Test Case Description	Test case procedure	Expected output	Result	Test Date
CREATE COURSE					
CC_1	Instructor creates course with empty name	1. Instructor clicks on "Thêm khóa học" from the menu. 2. Instructor inputs course name: "".	System shows an error message: "Tên khóa học không được bỏ trống".	Pass	20/09/2019
CC_2	Instructor creates course with name less than 6 characters	1. Instructor clicks on "Thêm khóa học" from the menu. 2. Instructor inputs course name: "Abc".	System shows an error message: "Tên khóa học không được ít hơn 6 kí tự".	Pass	20/09/2019
CC_3	Instructor creates course with name more than 255 characters	1. Instructor clicks on "Thêm khóa học" from the menu. 2. Instructor inputs course name: "*jasdhasdhjakshdkjdhaskdhaskuiowquiowqflkdsjfkdshfdskfhkjsdhfsjdkhfkjdshfkjhdskjfhkjsdhfkjhdjskhfkjdshkkfhsdjkfhksdfsdfsdffsdfsdffsdfsdfdsdfsdewrwerewrwerdglaksdlaposdwqesfjkjsldjaskldhwbrbkjxchaksdsdfsdfsafafdafsadfsdfsdfsdfsdfaewqw*".	System shows an error message: "Tên khóa học không được nhiều hơn 225 kí tự".	Pass	20/09/2019
CC_4	Instructor creates course with empty description	1. Instructor clicks on "Thêm khóa học" from the menu. 2. Instructor inputs course description: "".	System shows an error message: "Mô tả khóa học không được bỏ trống".	Pass	20/09/2019

CC_5	Instructor creates course with description less than 6 characters	1. Instructor clicks on "Thêm khóa học" from the menu. 2. Instructor inputs course description: "abc".	System shows an error message: "Mô tả khóa học không được ít hơn 6 kí tự".	Pass	20/09/2019
CC_6	Instructor creates course with wrong image format	1. Instructor clicks on "Thêm khóa học" from the menu. 2. Instructor inputs file "test.txt" for course image.	System shows an error message: "Hình ảnh không đúng định dạng".	Pass	20/09/2019
CC_7	Instructor create course with valid information	1. Instructor clicks on "Thêm khóa học" from the menu. 2. Instructor inputs valid information.	System creates course and shows the message "Tạo khóa học thành công". System redirects to course details page with empty content	Pass	20/09/2019
CREATE INTERACTIVE LESSON					
IL_1	Instructor creates interactive lesson with invalid information	1. Instructor clicks on "Thêm trận đấu" from the course details page. 2. Instructor inputs invalid information: • Lesson name is empty, less than 6 or more than 225 characters • Description is empty or less than 6 characters	System shows error messages for each error.	Pass	20/09/2019
IL_2	Instructor creates interactive lesson with empty move list	1. Instructor clicks on "Thêm trận đấu" from the course details page. 2. Instructor does not perform any move.	The preview button is disabled.	Pass	20/09/2019

IL_3	Instructor creates depth 2 move	1. Instructor clicks on "Thêm trận đấu" from the course details page. 2. Instructor performs: "e4, e5, d4, d5" 3. Instructor selects e5 move 4. Instructor perform c4 move 5. Instructor confirms to add another move	C4 move is added successfully. It is the second move and has same previous position with d4 move.	Pass	20/09/2019
IL_4	Instructor edits move	1. Instructor clicks on "Thêm trận đấu" from the course details page. 2. Instructor performs: "e4, e5, d4, d5" 3. Instructor selects e5 move 4. Instructor perform c4 move 5. Instructor confirms to edit move	Lesson content is change to "e4, e5, c4. D5 move is removed because c4 move will start new position.	Pass	20/09/2019
IL_5	Instructor add content for move	1. Instructor clicks on "Thêm trận đấu" from the course details page. 2. Instructor performs: "e4, e5, d4, d5" 3. Instructor selects e5 move 4. Instructor add content "abcxyz" for e5 move	New lesson content is added for e5 move.	Pass	20/09/2019
IL_6	Instructor create interactive lesson with valid information and non-empty move list	1. Instructor clicks on "Thêm trận đấu" from the course details page. 2. Instructor performs: "e4, e5, d4, d5" 3. Instructor clicks on "Xem trước" button 4. Instructor clicks on "Lưu" button	New interactive lesson is added and system shows a message "Tạo trận đấu thành công".	Pass	20/09/2019

CREATE UNINTERACTIVE LESSON					
UL_1	Instructor create uninteractive lesson with invalid information	1. Instructor clicks on "Thêm bài đọc" from the course details page. 2. Instructor inputs invalid information: • Lesson name is empty, less than 6 or more than 225 characters • Description is empty or less than 6 characters	System shows error messages for each error.	Pass	20/09/2019
UL_2	Instructor create uninteractive lesson with valid information	1. Instructor clicks on "Thêm bài đọc" from the course details page. 2. Instructor inputs invalid information: • Lesson name: Bài đọc test • Description: Bài đọc test case • Nội dung: Bàn cờ vua gồm có 64 ô cờ	New uninteractive lesson is added and system shows a message "Tạo bài đọc thành công".	Pass	20/09/2019
CREATE EXERCISE					
EX_1	Instructor create exercise with invalid information	1. Instructor clicks on "Thêm bài đọc" from the course details page. 2. Instructor inputs invalid information: • Exercise name is empty, less than 6 or more than 225 characters • Question is empty or less than 6 characters • Description is empty or less than 6 characters	System shows error messages for each error.	Pass	20/09/2019
EX_2	Instructor create exercise with valid manual exercise	1. Instructor clicks on "Thêm bài tập" from the course details page. 2. Instructor clicks on the edit chess position	New manual exercise is added and system shows a message "Tạo bài tập thành công".	Pass	20/09/2019

		3. Instructor drags pieces and confirms new chess position: Ka1, Qa2 (White), Kb7, Qa5 (Black) 4. Instructor performs move: Qxa5 5. Instructor clicks on "Xem trước" button 6. Instructor clicks on "Lưu" button			
EX_3	Instructor create exercise with valid bot exercise	1. Instructor clicks on "Thêm bài tập" from the course details page. 2. Instructor clicks on the edit chess position 3. Instructor drags pieces and confirms new chess position: Ka1, Qa2 (White), Kb7 (Black) 4. Instructor selects bot answer 5. Instructor clicks on "Xem trước" button 6. Instructor clicks on "Lưu" button	New manual exercise is added and system shows a message "Tạo bài tập thành công".	Pass	20/09/2019
EDIT COURSE STATUS					
CS_1	Instructor submit drafting course	1. Instructor clicks on the setting button. 2. Instructor clicks on the submit button	Course status is changed to Waiting.	Pass	20/09/2019

Table 109: Test cases <Instructor>

4.1.5 <Admin> Review New Course – Review Instructor - Create Category

ID	Test Case Description	Test case procedure	Expected output	Result	Test Date
REVIEW NEW COURSE					

RC_1	Admin approves new course	1. Admin clicks on "Quản lý khóa học" 2. Admin selects course 3. Admin clicks "Đồng ý" button	System shows messages: "Khóa học đã được công khai".	Pass	20/09/2019
RC_2	Admin rejects new course	1. Admin clicks "Quản lý khóa học" 2. Admin clicks course 3. Admin clicks "Từ chối" button	System shows messages: "Khóa học đã bị từ chối công khai".	Pass	20/09/2019
REVIEW INSTRUCTOR					
RI_1	Admin approves new instructor	1. Admin clicks "Xét duyệt giảng viên" 2. Admin clicks new instructor 3. Admin clicks "Đồng ý" button	System shows messages: "Người hướng dẫn đã được chấp nhận".	Pass	20/09/2019
RI_2	Admin reject new instructor	1. Admin clicks "Xét duyệt giảng viên" 2. Admin clicks new instructor 3. Admin clicks "Từ chối" button 4. Admin inputs rejected reason and confirm	System sends email with rejected reason for the user.	Pass	20/09/2019
CREATE CATEGORY					
CA_1	Admin create category with empty name	1. Admin clicks "Thêm danh mục". 2. Admin inputs category name: "" 3. Admin clicks "Đồng ý" button.	System shows error messages:"Tên danh mục không được bỏ trống".	Pass	20/09/2019
CA_2	Admin create category with name less than 6 characters	1. Admin clicks "Thêm danh mục". 2. Admin inputs category name: "abc" 3. Admin clicks "Đồng ý" button.	System shows error messages:"Tên danh mục không được ít hơn 6 kí tự".	Pass	20/09/2019
CA_3	Admin create category with name	1. Admin clicks "Thêm danh mục". 2. Admin inputs category name: "*jasdhasdhjakshdkjdhaskdhaskuiowquio*	System shows error messages:"Tên danh mục không được nhiều hơn 225 kí tự".	Pass	20/09/2019

	more than 225 characters	*wqflkdsjfkdshfdskfhkjsdhfsjdkhfkjdshfkjhdskjfhkjsdhfkjhdjskhfkjdshkkfhsdjkfhksdfsdfsdffsdfsdffsdfsdfdsdfsdewrwerewrwerdglaksdlaposdwqesfjkjsldjaskldhwbrbkjxchaksdsdfsdfsafafdafsadfsdfsdfsdfsdfaewqw"* 3. Admin clicks "Đồng ý" button.			
CA_4	Admin create category with valid information	1. Admin clicks "Thêm danh mục". 2. Admin inputs category name: "Khai cuộc" 3. Admin clicks "Đồng ý" button.	System shows the message "Tạo danh mục thành công" and a new category is added.	Pass	20/09/2019

Table 110: Test cases <Admin>

F. Software User's Manual

1. Installation Guide

1.1 Setting up environment at server side

1.1.1 Hardware requirements

Server	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (8 Mbps)	Cable, Wi-Fi (50 Mbps)
Operating System	Window Server 2008	Window Server 2012
Computer Processor	Intel® Xeon ® 1.4GHz	Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz)
Computer Memory	4GB RAM	32GB RAM or more
Storage Space	4GB	8GB or more

Table 111: Hardware Requirement for Server

1.1.2 Software requirements

Name	Name/Version	Description
Environment	Java JDK 8, Node v10	Specification for developing web server
Package manager	npm v6	Package manager for JavaScript Programming
Operating System	Window 10	Operating system and platform for development
Integrated development environment	- IntelliJ IDEA 2019.1.1 - Visual Studio Code	Programming tools
Modeling tool	draw.io, lucidchart	Software modeling tool
Database Management System	MySQL 5.7	Used to create & manage the database for system
Temporary Storage	Redis (2.8.6 or higher version)	Used to store data temporarily and load balancing data of MySQL
Source control	Github.com	Used to source control & version control.
Web browser	Chrome 76	Testing browser

Table 112: Software Requirement for Server.

1.2 Deployment

1.2.1 Prepare development package

Step 1: Download and Install Java 8

Link download:
http://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8-2177648.html

Step 2: Download and setup Apache Tomcat 8

Link download: https://tomcat.apache.org/download-80.cgi

Step 3: Download and Install Node v10

Link download: https://nodejs.org/en/download/

Step 4: Download and Install Redis v3.2.1

Link download: https://github.com/microsoftarchive/redis/releases

Step 5: Download and Install IntelliJ IDE

Link download: https://www.jetbrains.com/idea/download

Step 6: Download, Install and configure MySQL

Link download: https://dev.mysql.com/downloads/installer/

1.2.2 Configuration of Web Service / Web Application

1.2.2.1 Confirgure Web Service before deploy

Step 1: Go to file application.properties

Step 2: Input datasource information

spring.datasource.url=jdbc:sqlserver://**server_name**;databaseName=**db_name**
spring.datasource.username=**username**
spring.datasource.password=**password**

Step 3: Input redis datasource information (Without any redis configuration, the system will automatically connect to local under port 6379)

spring.redis.host= **host**

spring.redis.port = **port**

spring.redis.password = **password** (if redis is security)

Step 4: Save file

1.2.2.2 Configure Web Application before deploy

Step 1: Go to https://console.firebase.google.com

Step 2: Create an Web application and get Firebase SDK Config

Step 3: Go to file main.js

Step 4: Input firebase config

Step 5: Save file

1.2.3 Deployment of Web Service / Web application

1.2.3.1 Deploy MySQL

Register Amazon Web Services Account in https://aws.amazon.com

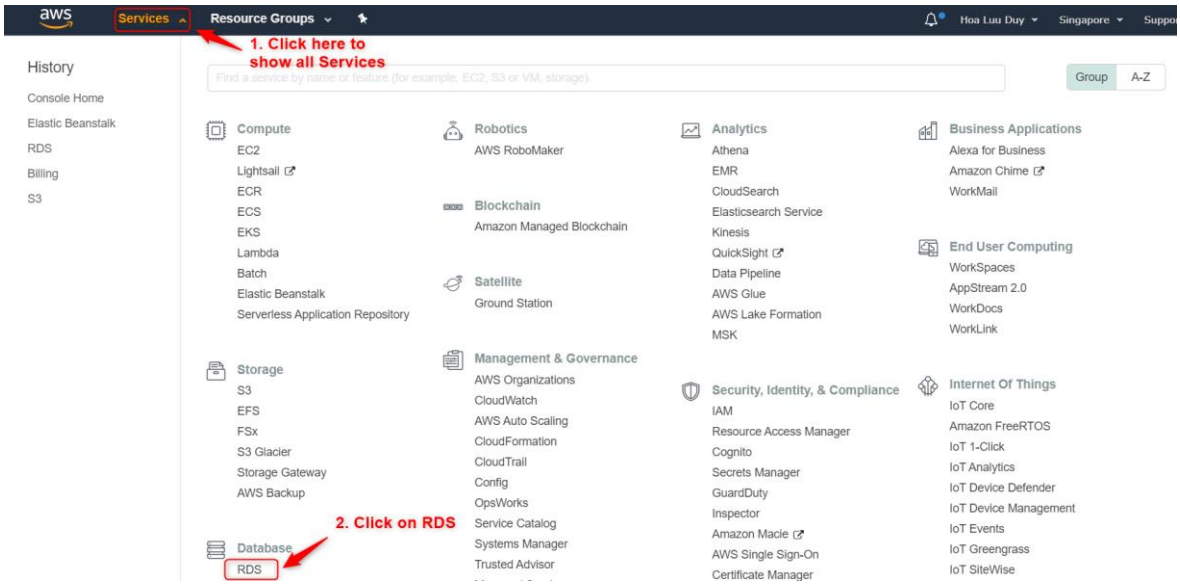


Figure 116: <MySQL> Select Service

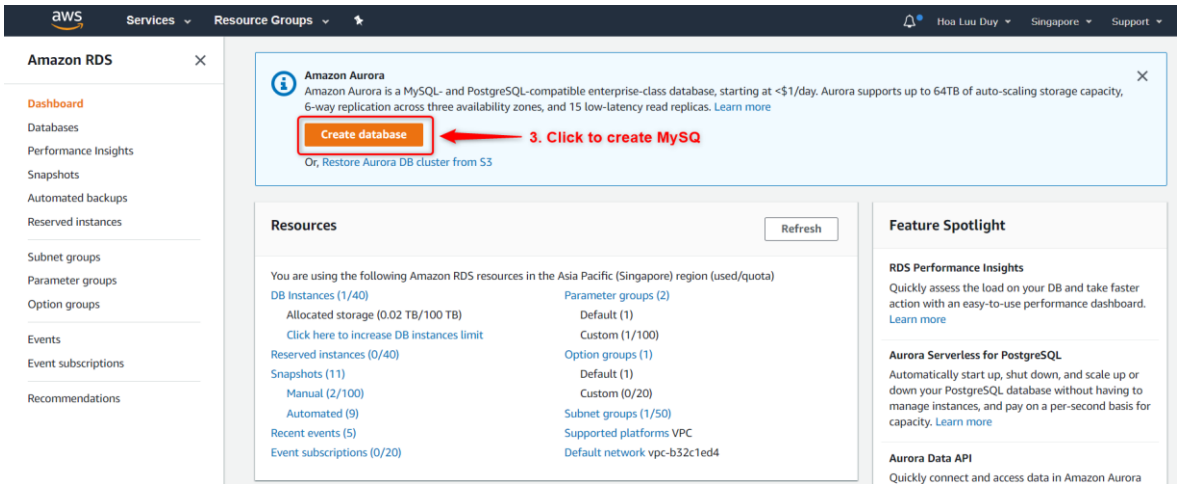


Figure 117: <MySQL> Create Database

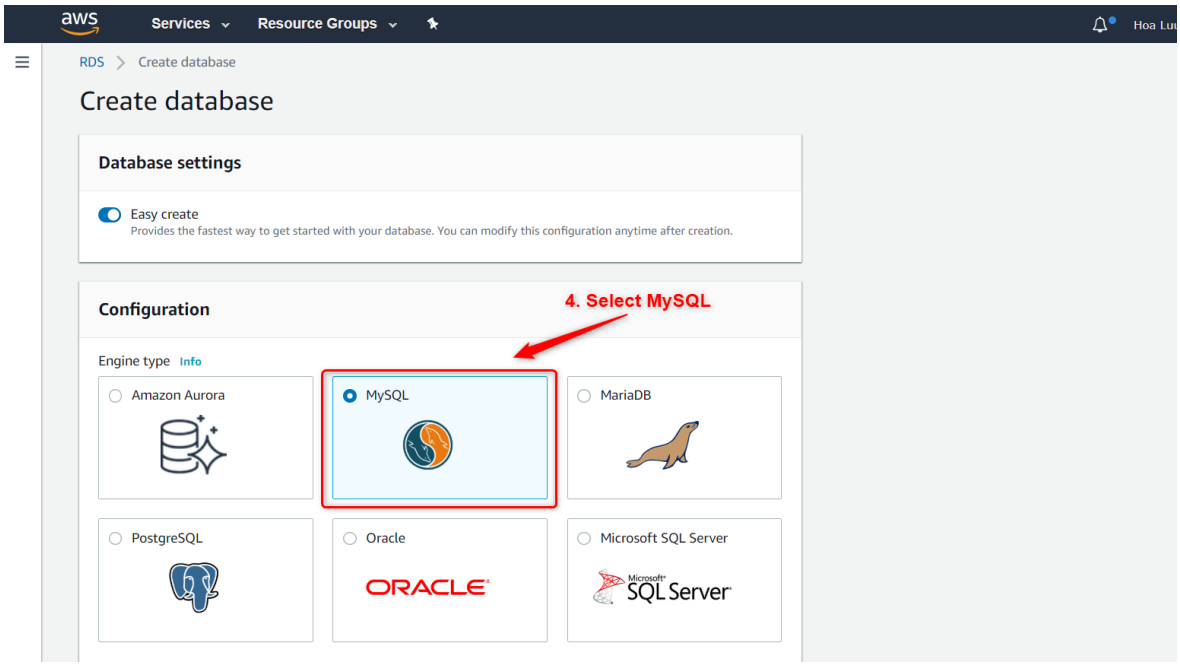


Figure 118: <MySQL> Select Database

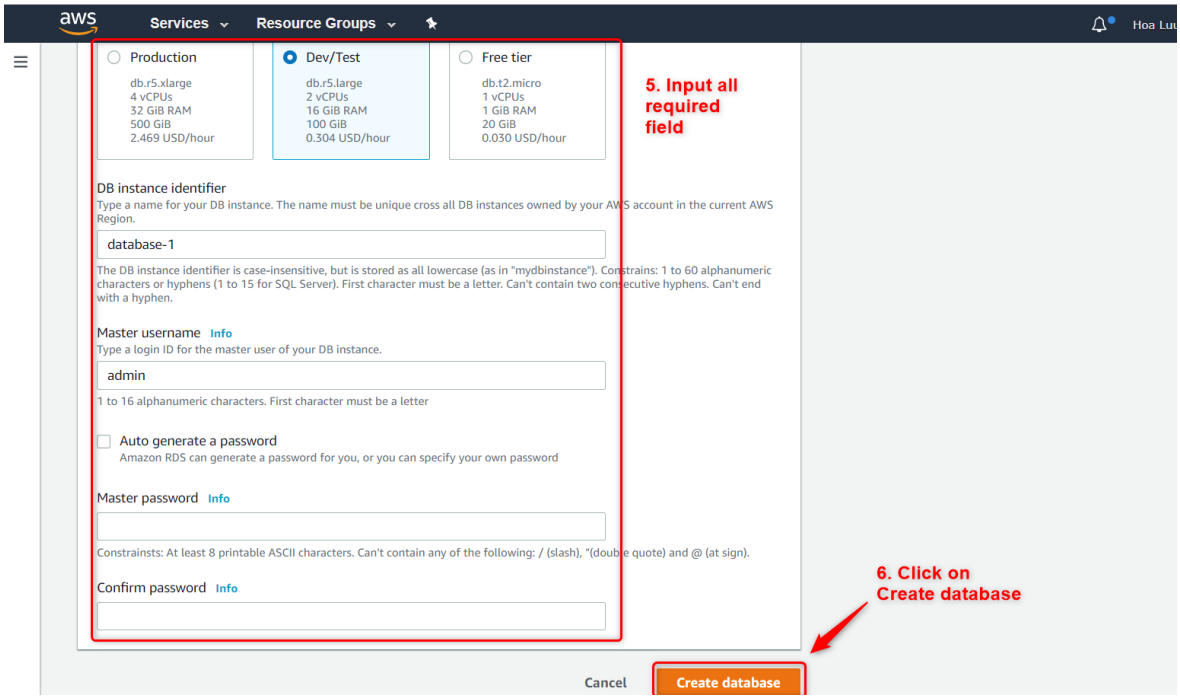


Figure 119: Input Information

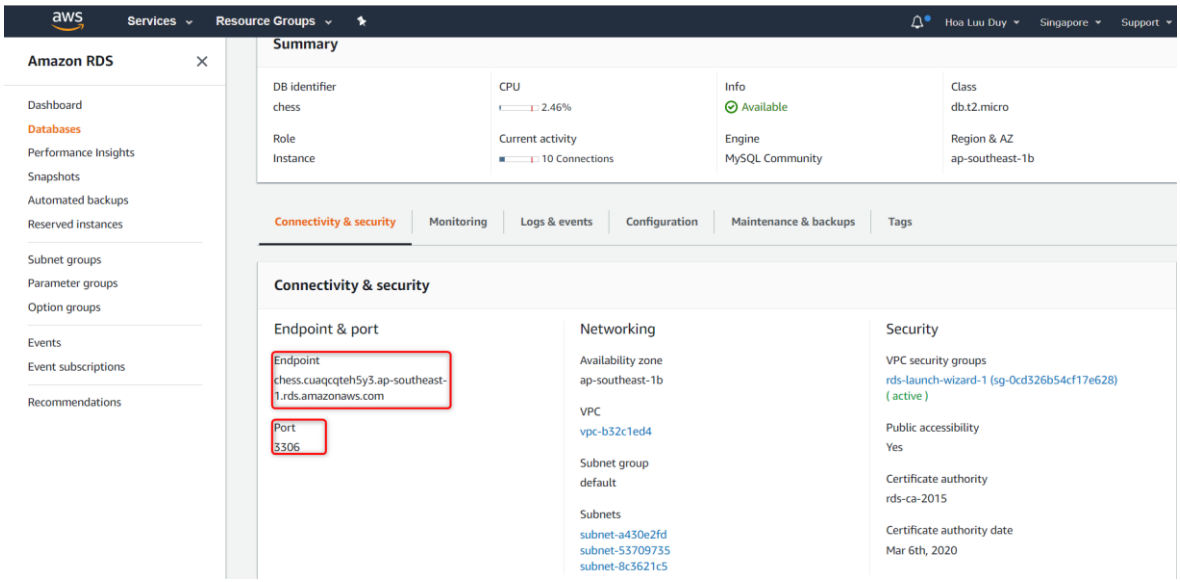


Figure 120: <MySQL> Configuration

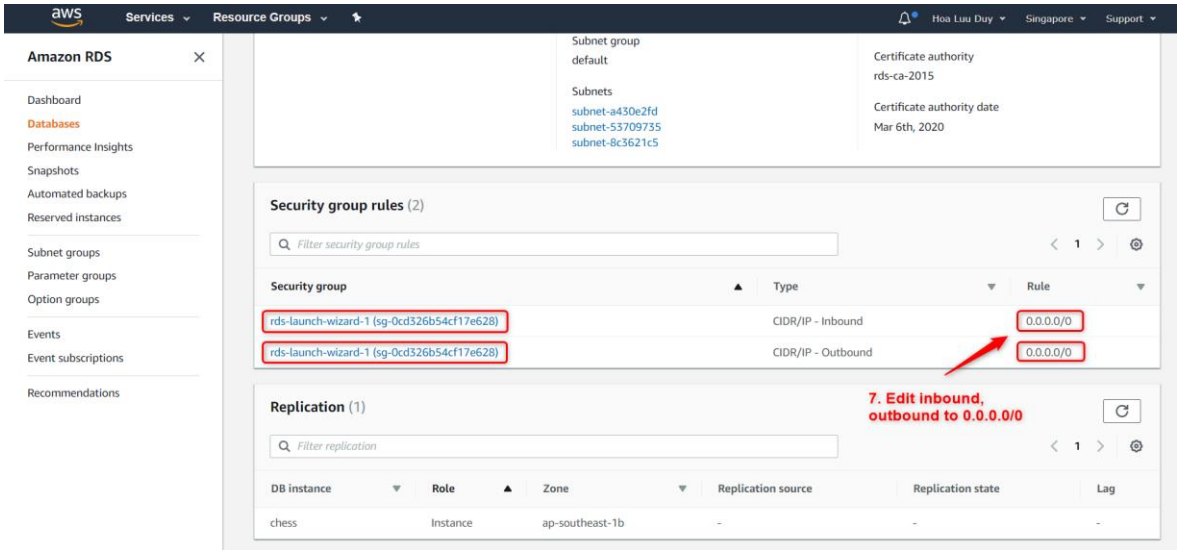


Figure 121: <MySQL> Edit Rule

Open MySQL Workbench 8.0 CE. Create new connection by following:

- Hostname: Endpoint of MySQL in AWS
- Port: Port of MySQL in AWS
- Username: Username of MySQL in AWS.
- Password: password of the above username.

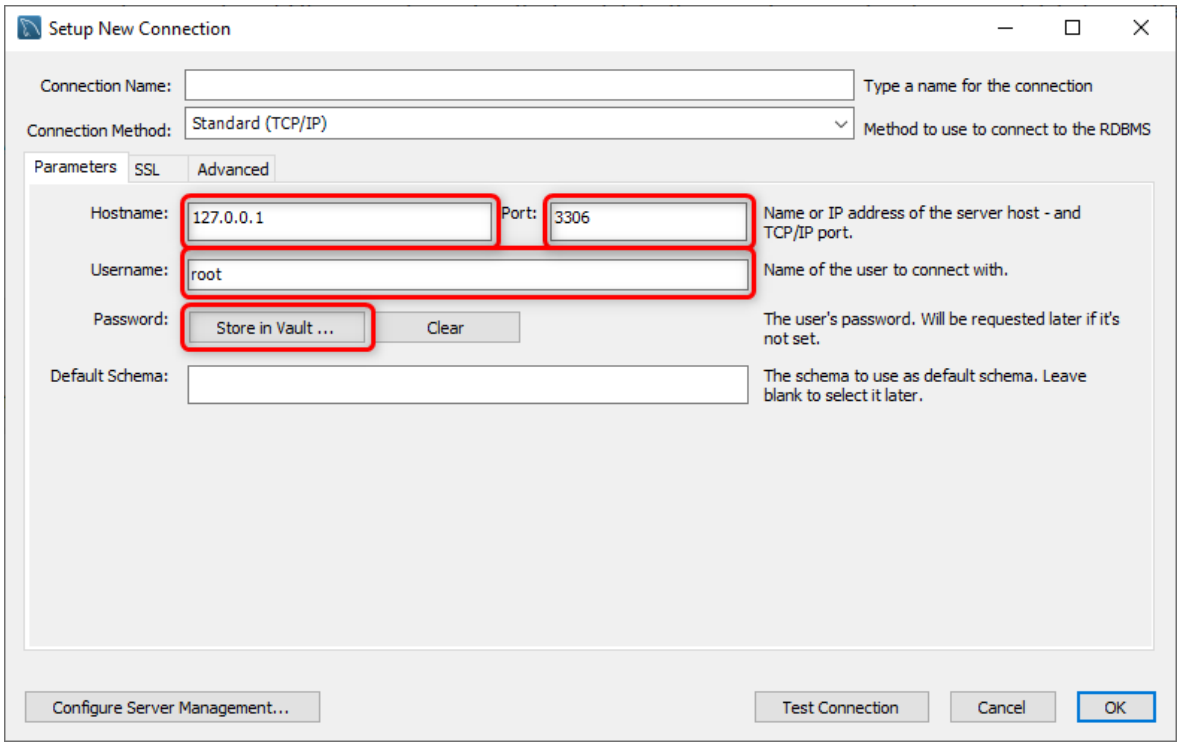


Figure 122: <MySQL> Using in Local

- Click connect and wait for connection to be completed

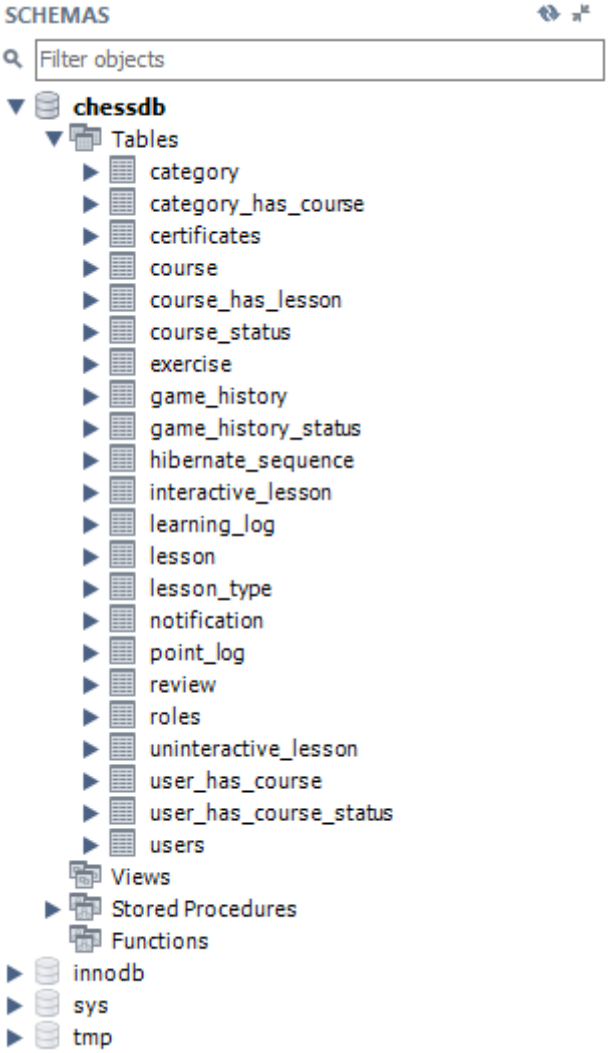


Figure 123: COLS Database

1.2.3.2. Deploy Web server

Register Amazon Web Services Account in https://aws.amazon.com

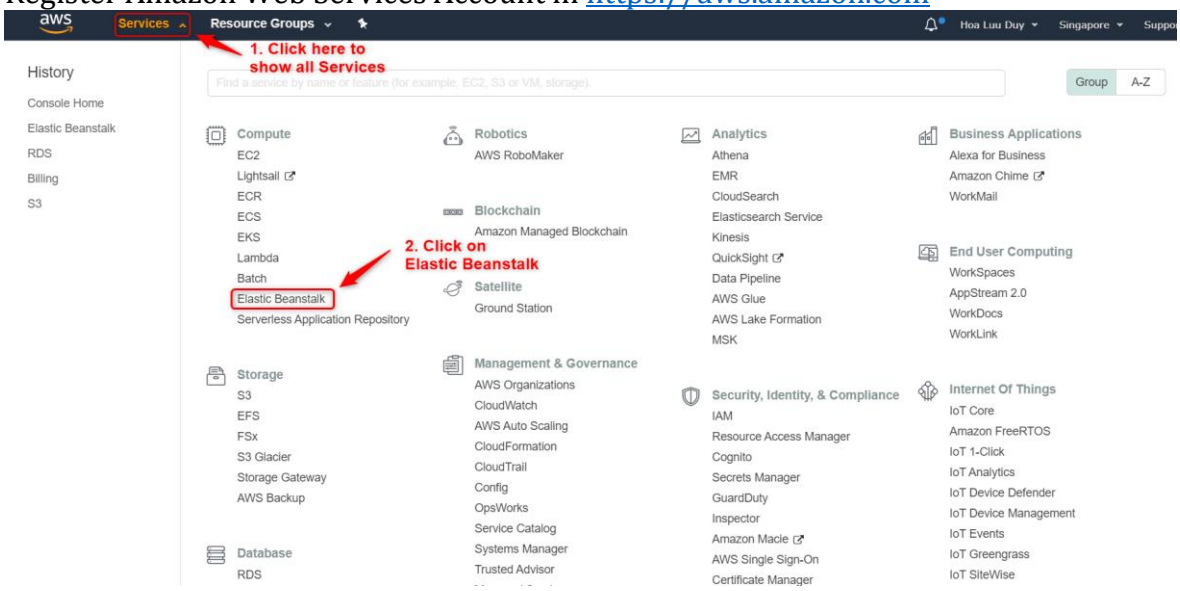


Figure 124: <Web Server> Select Service

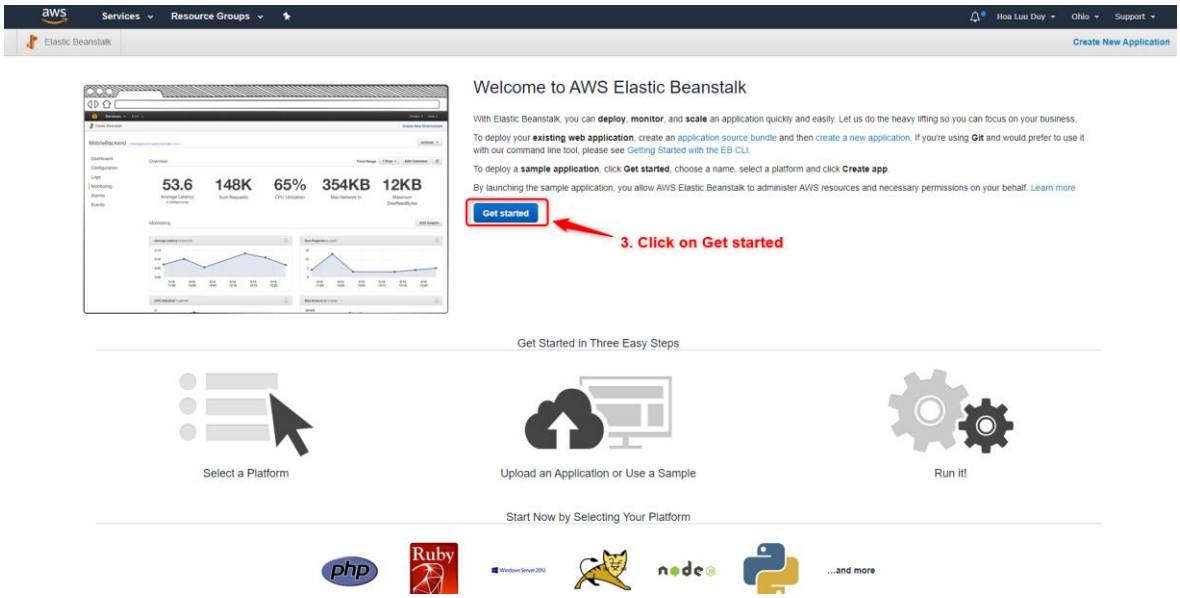


Figure 125: <Web Server> Start Elastic Beanstalk

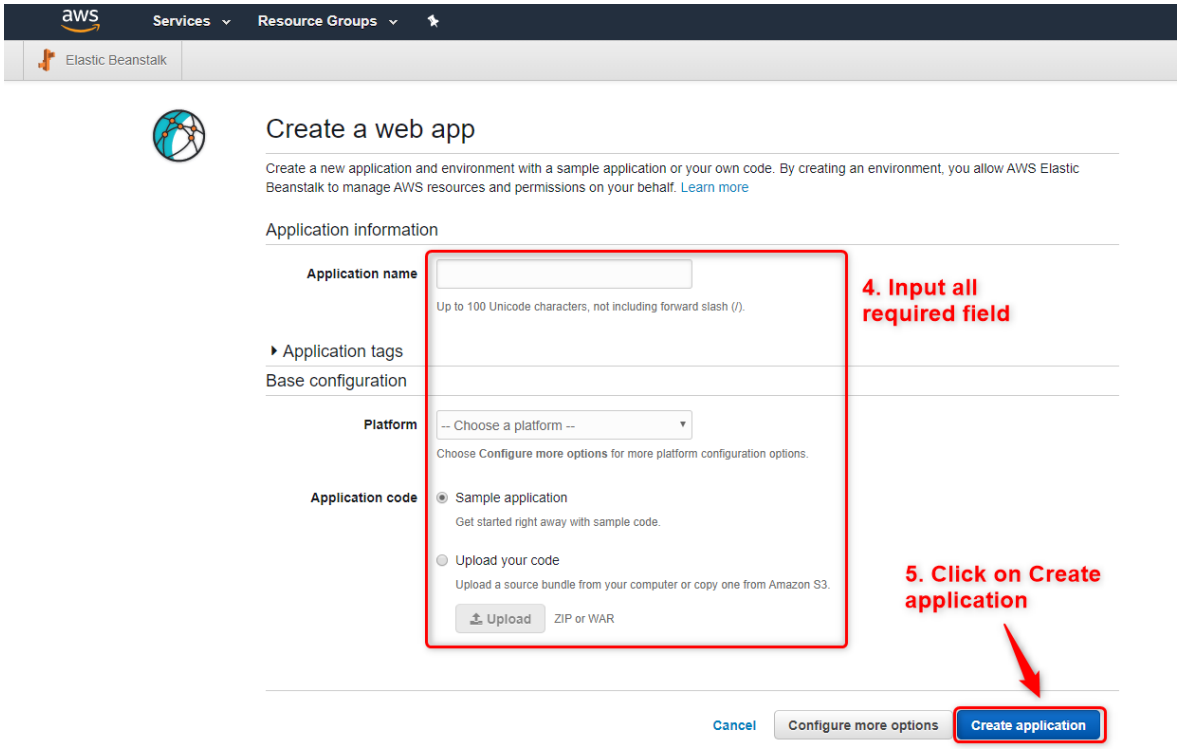


Figure 126: <Web Server> Input Information

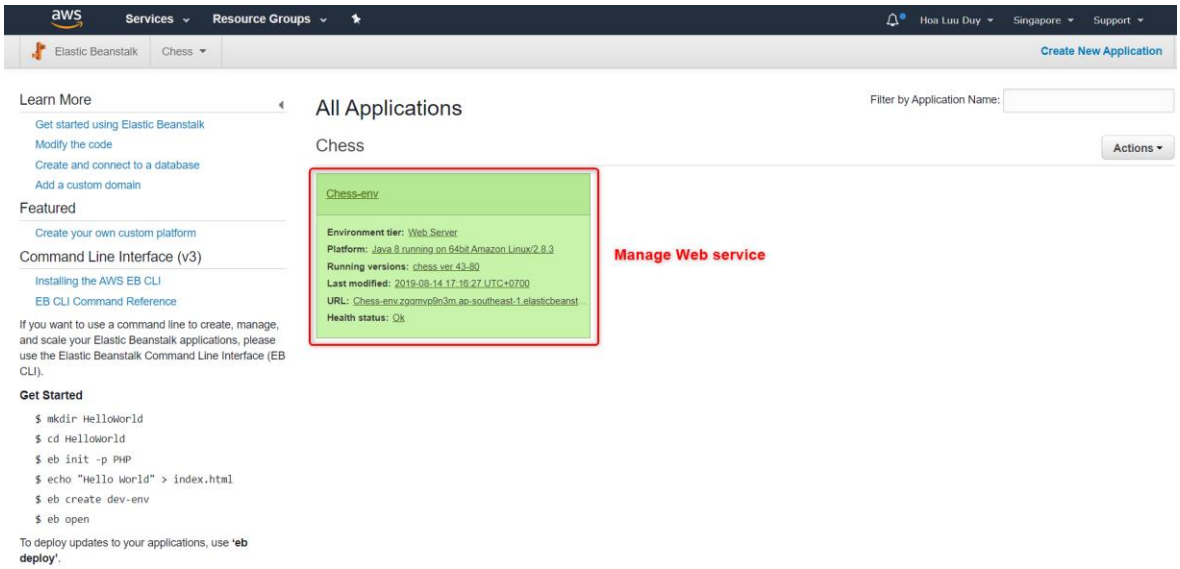


Figure 127: <Web Server> Manage Web Service

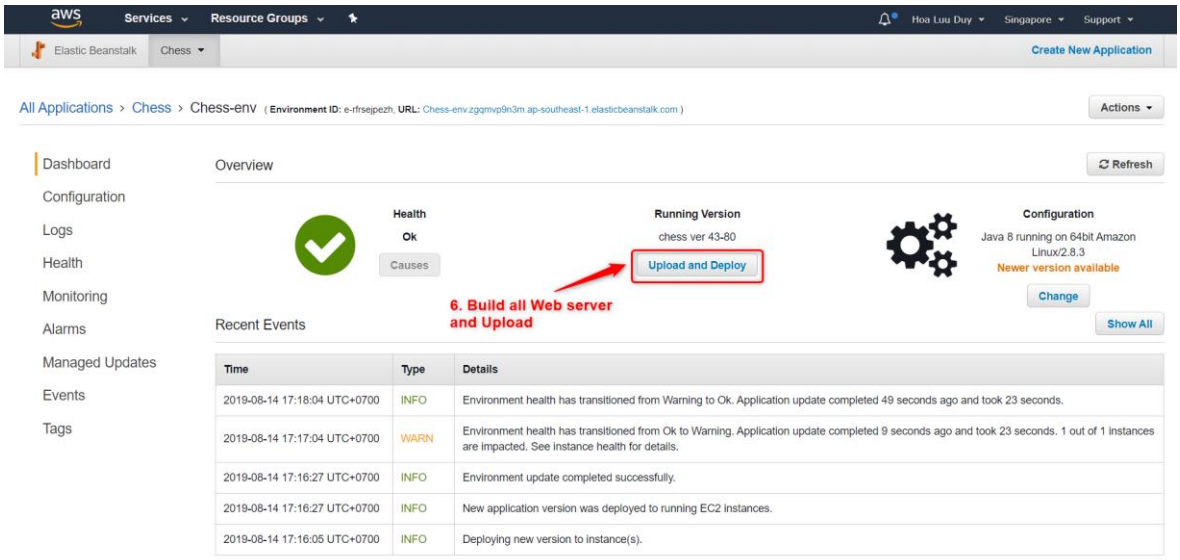


Figure 128: <Web Server> Upload File

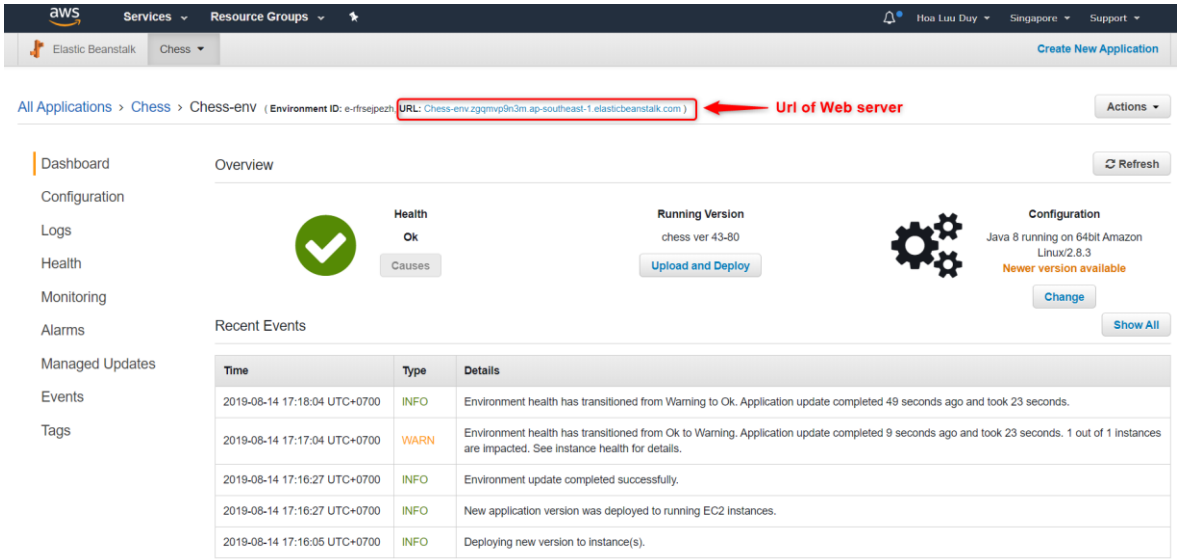


Figure 129: <Web Server> Result

1.2.3.3 Deploy Web Application

Register Amazon Web Services Account in https://aws.amazon.com

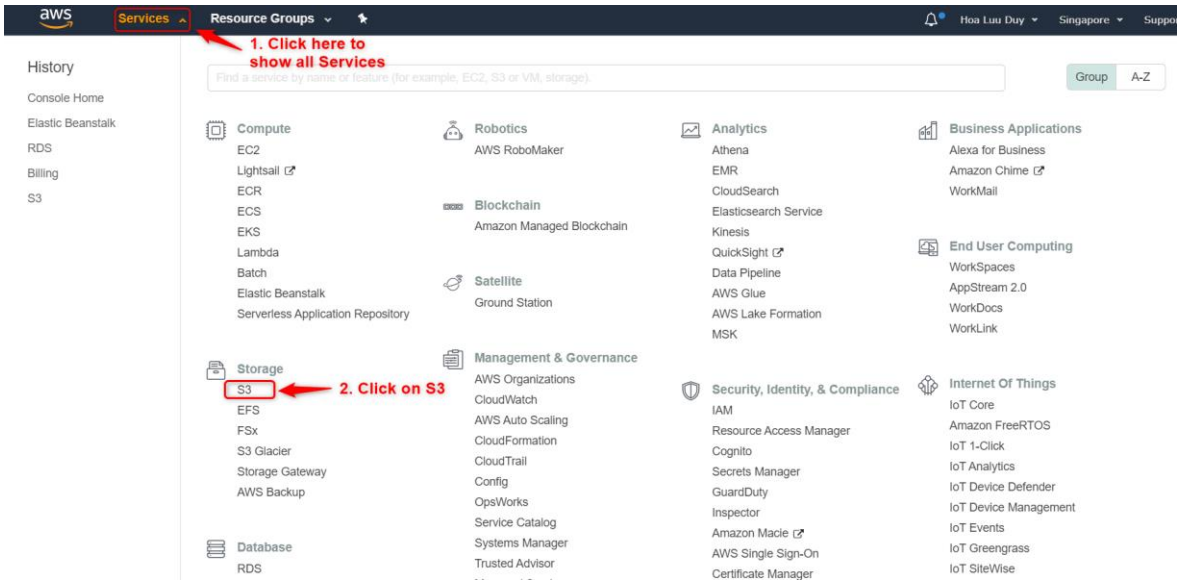


Figure 130: <Front-end App> Select Service

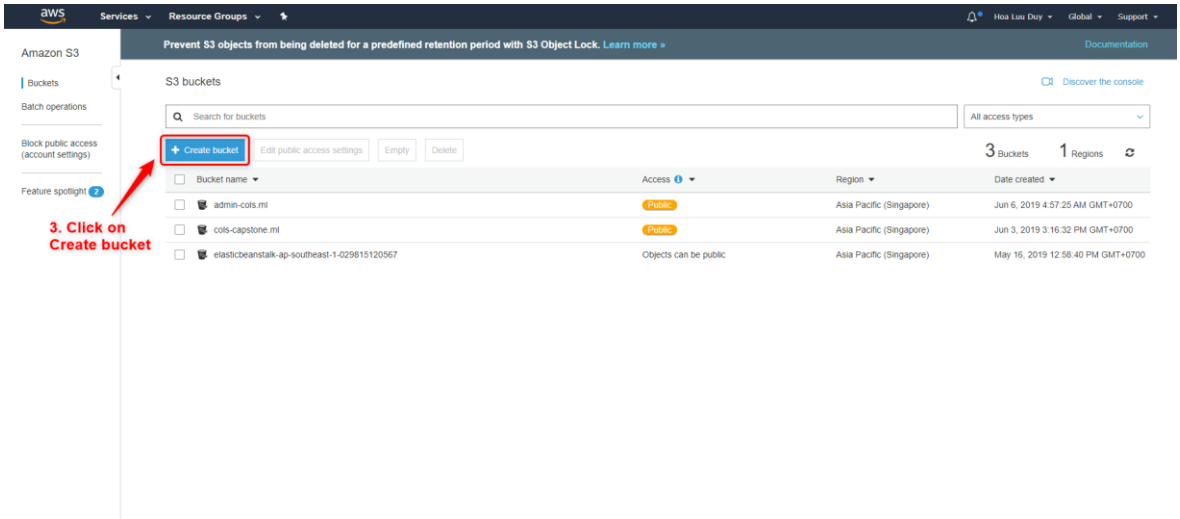


Figure 131: <Front-end App> Create Bucket

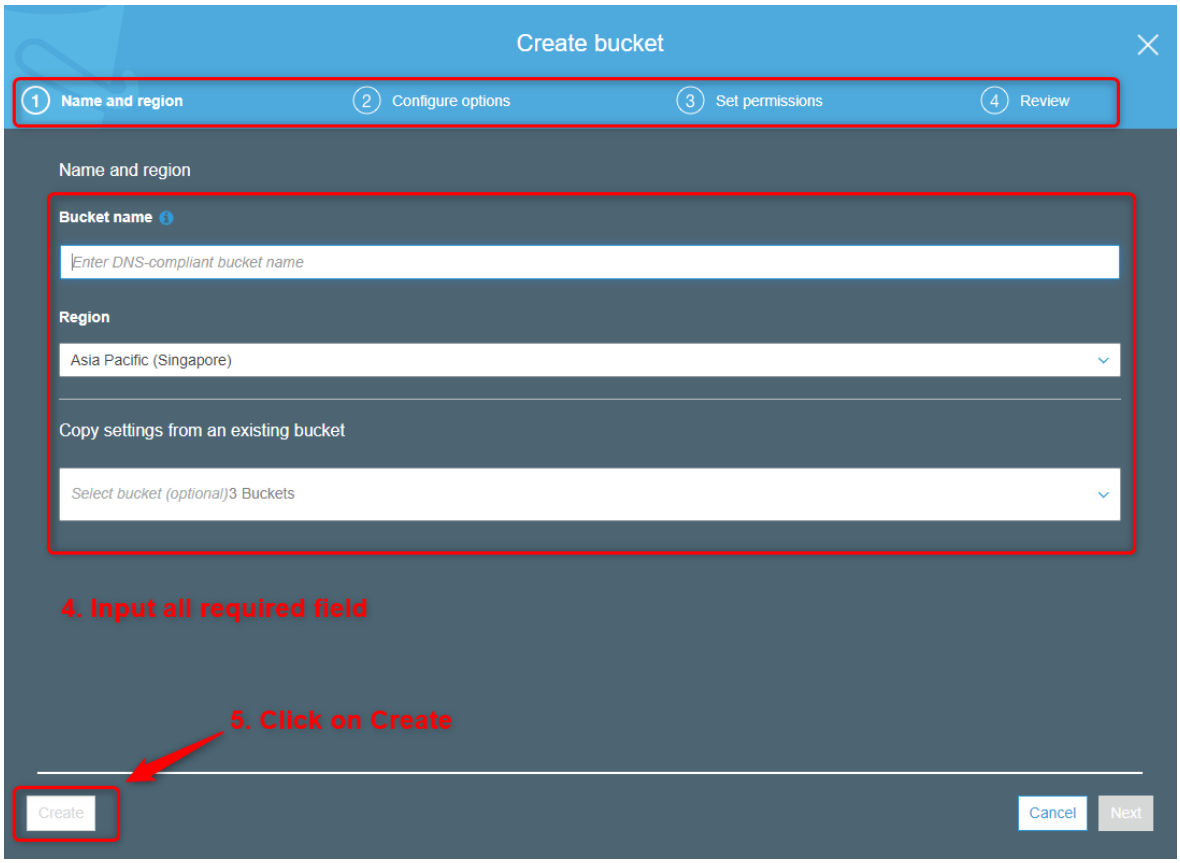


Figure 132: <Front-end App> Input Information

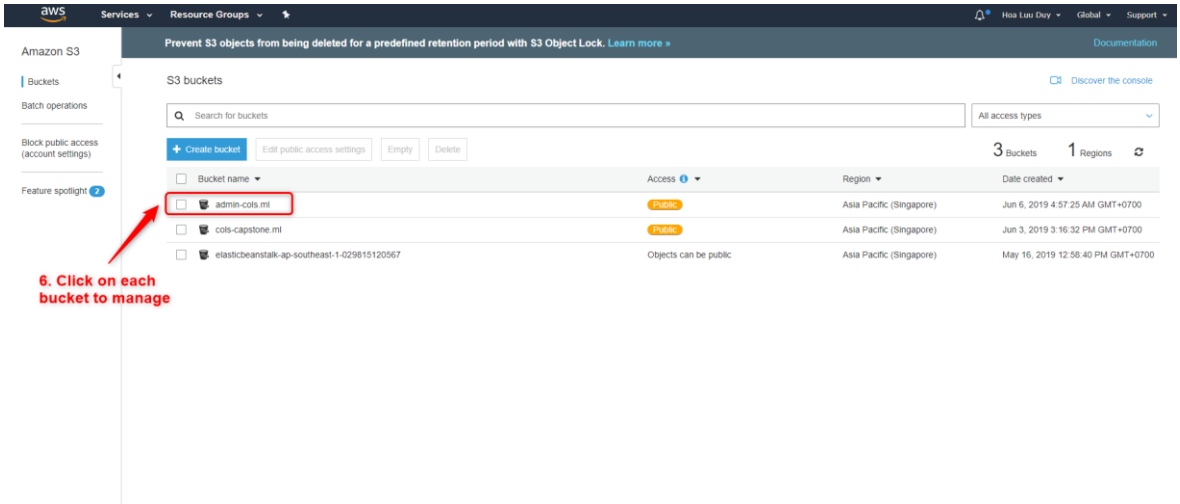


Figure 133: <Front-end> App Navigate To a bucket

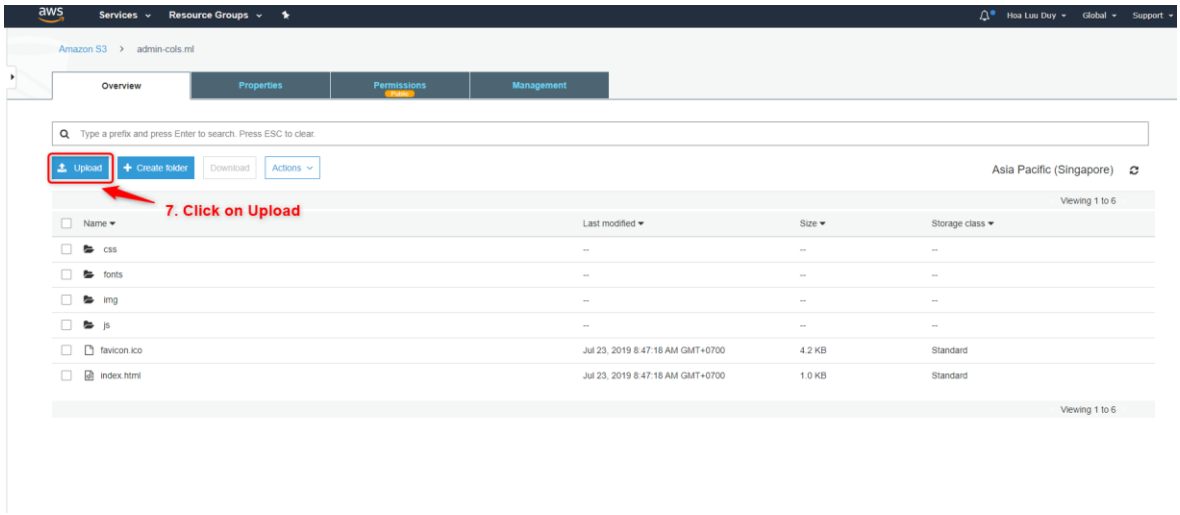


Figure 134: <Front-end App> Select Upload

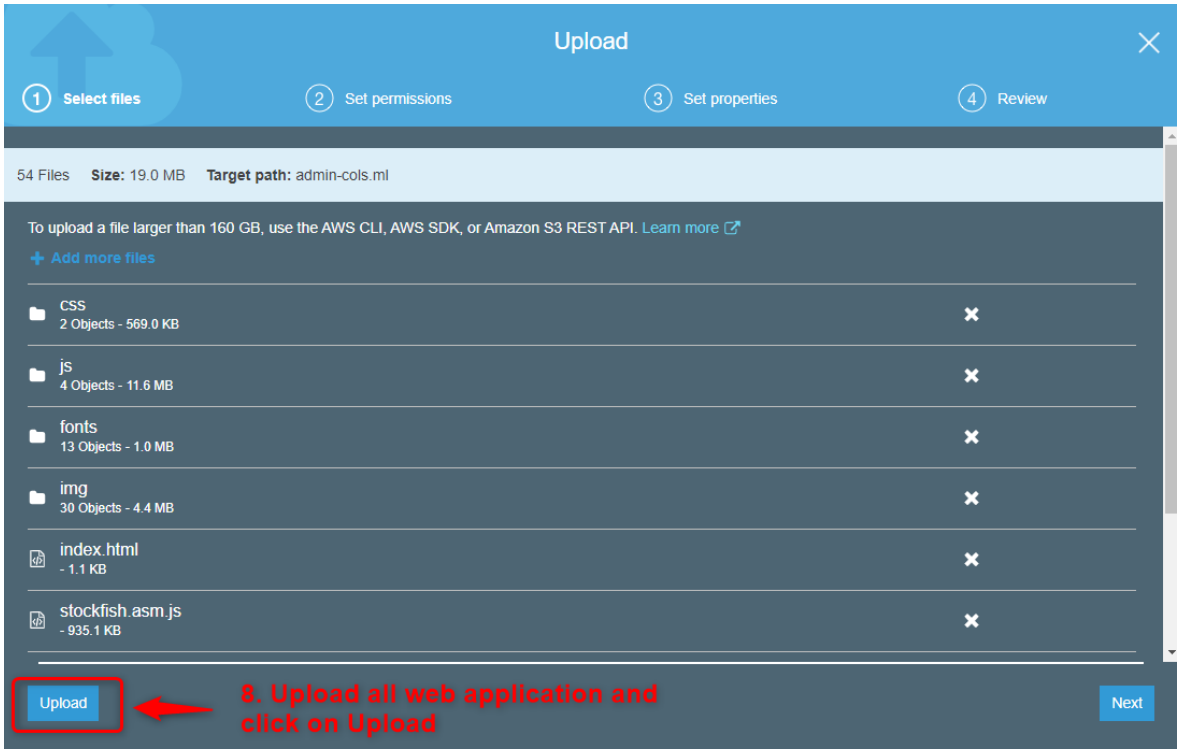


Figure 135: <Front-end App> Input Information and Click On Upload Button

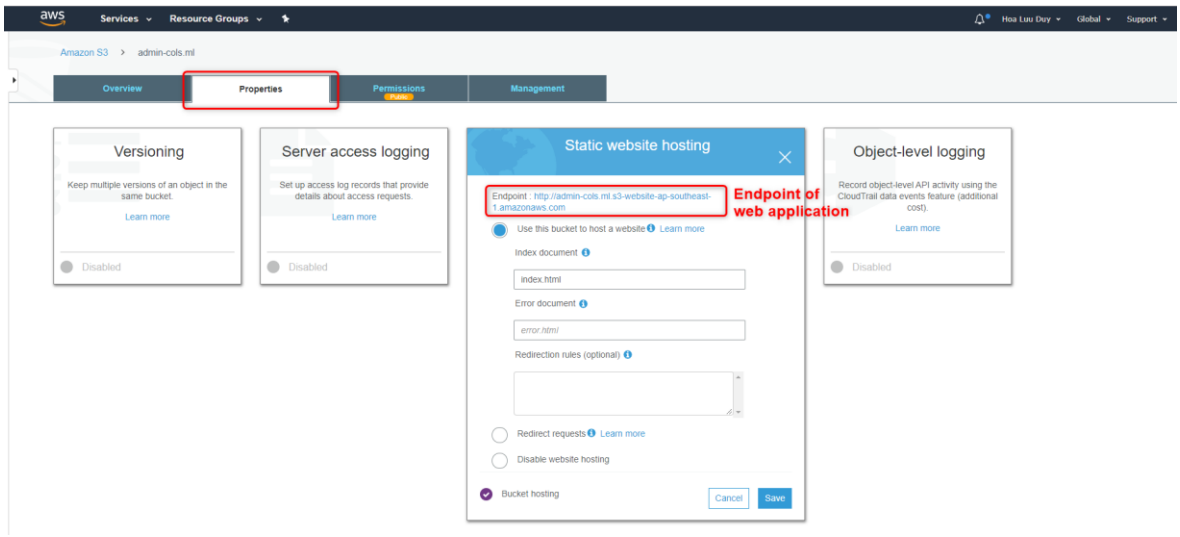


Figure 136: <Front-end App> Configuration

2. User guide

2.1 Guest

2.1.1 Search Course

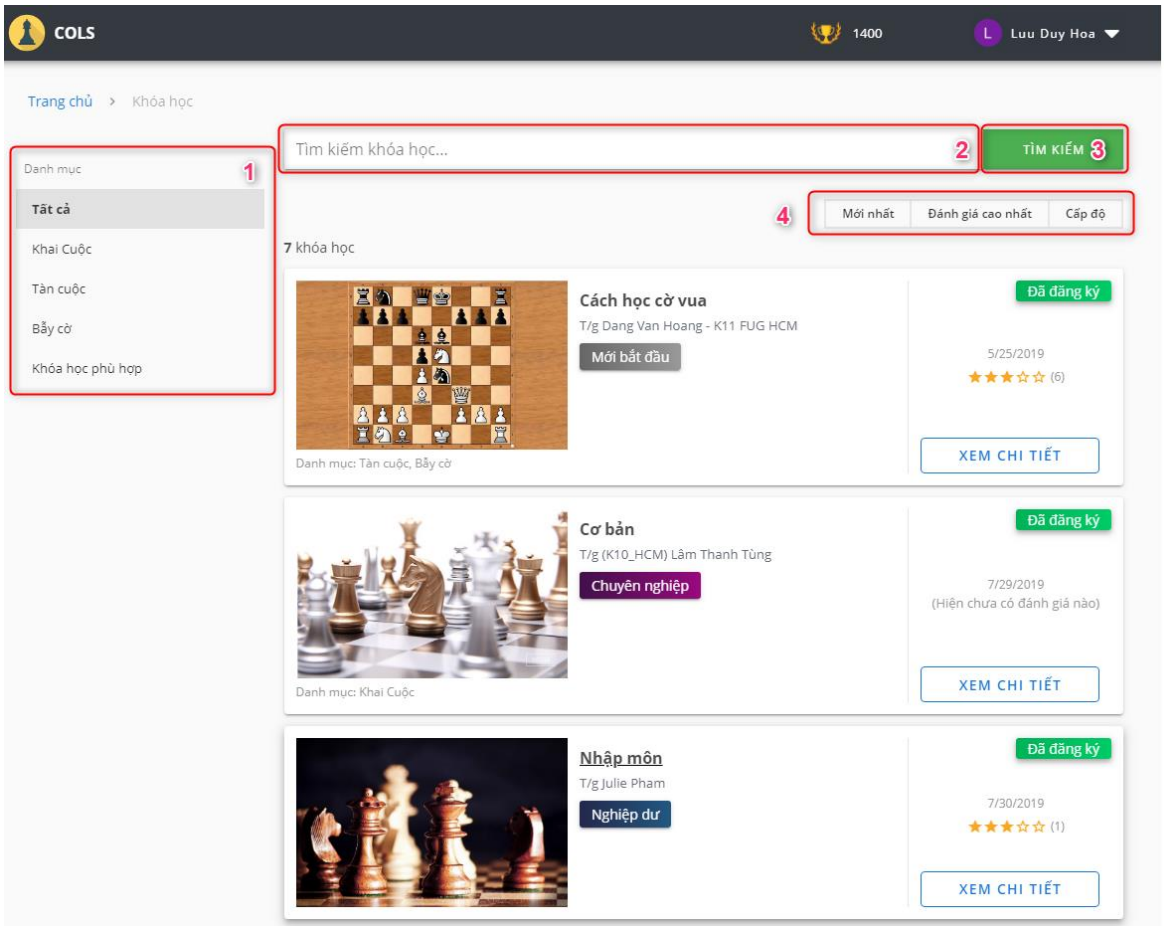


Figure 137: Search Course

Step	Description
1	Click on “Bắt đầu học”

2	Select Category to search
3	Type keyword to search
4	Select sort type
5	Click on “Tìm kiếm” to search
6	Click on the title, image, button to view detail

Table 113: Search Course description

2.1.2 View Course Details

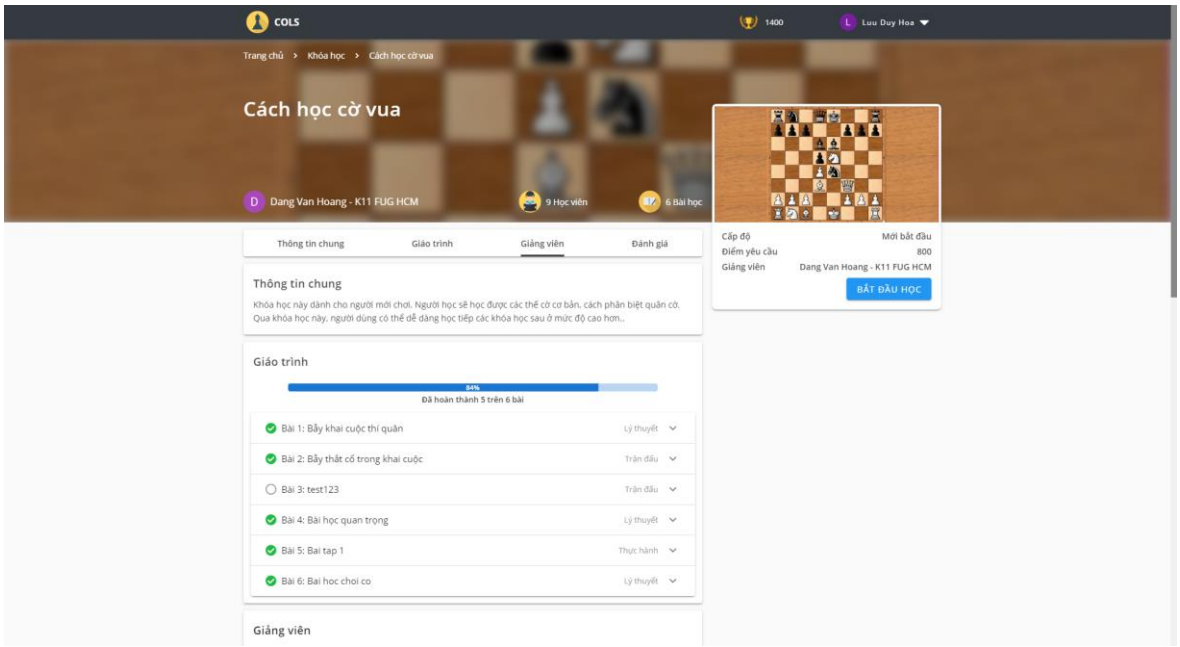


Figure 138: View Course Details

Step	Description
1	After list course are show, click on each course to see detail

Table 114: View Course Details description

2.2 Learner

2.2.1 Enrol Course

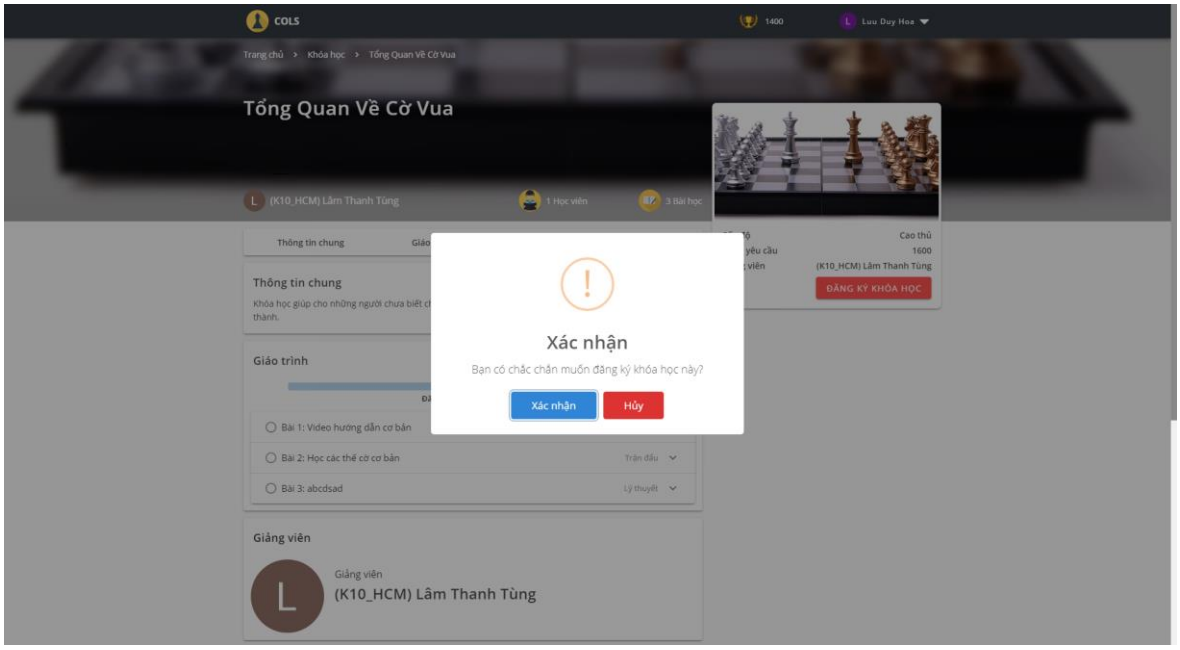


Figure 139: Enrol Course

Step	Description
Pre-condition	Login as role Learner
1	Click on each course to show detail
2	Click on “Đăng ký khóa học”
3	Click on “Xác nhận”

Table 115: Enrol Course description

2.2.2 Learn Lesson

Figure 140: Learn Lesson

Step	Description
Pre-condition	Login as role Learner. Enrol to course.
1	Click on each course to show detail
2	Click on “Bắt đầu học”

Table 116: Learn Lesson description

2.2.3 Create Review

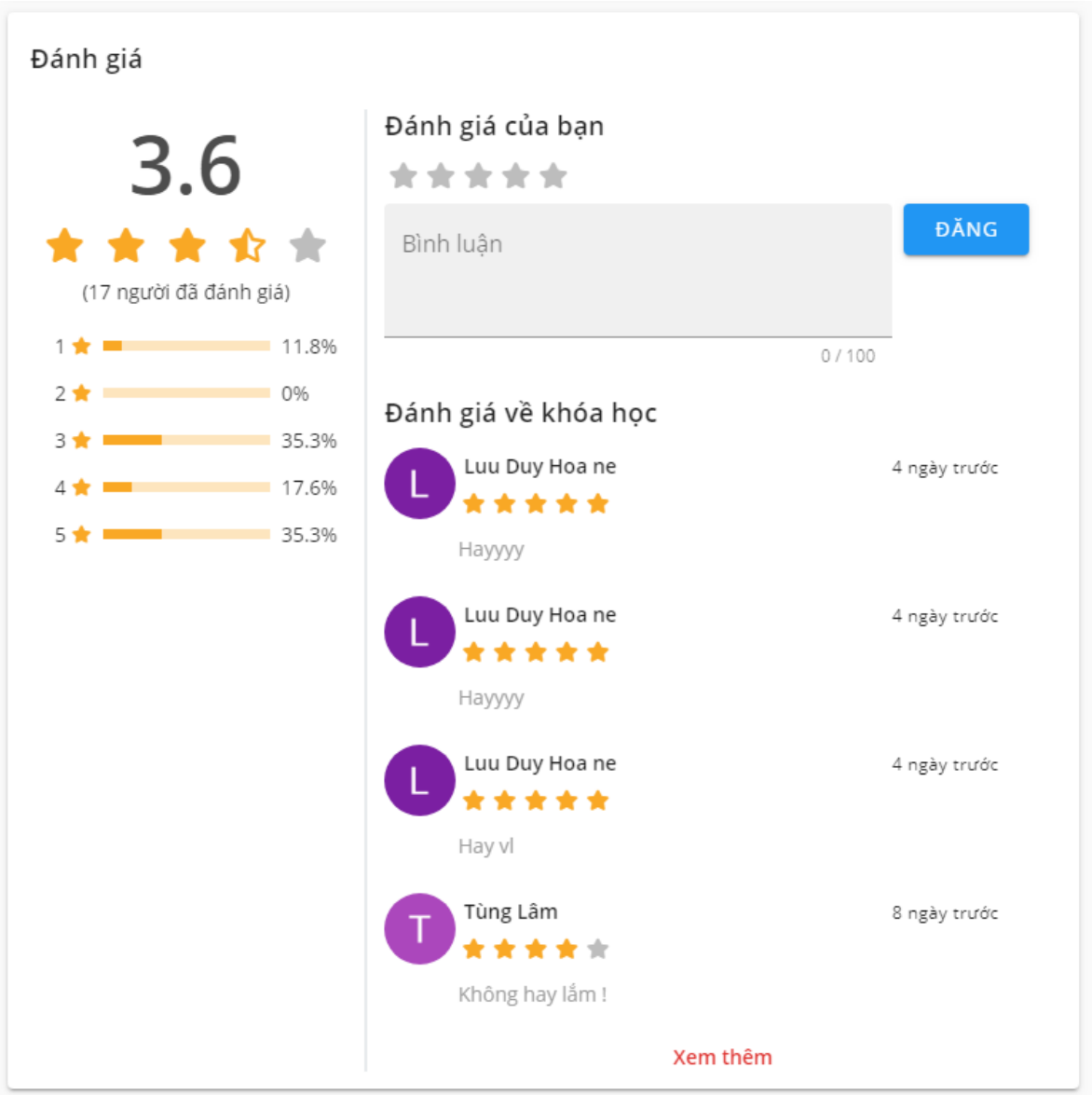


Figure 141: Create review

Step	Description
Pre-condition	Login as role Learner. Enrol to course.
1	Click on each course to show detail
2	After fill all field, click on "Đăng"

Table 117: Create Review description

2.2.4 Manage Review

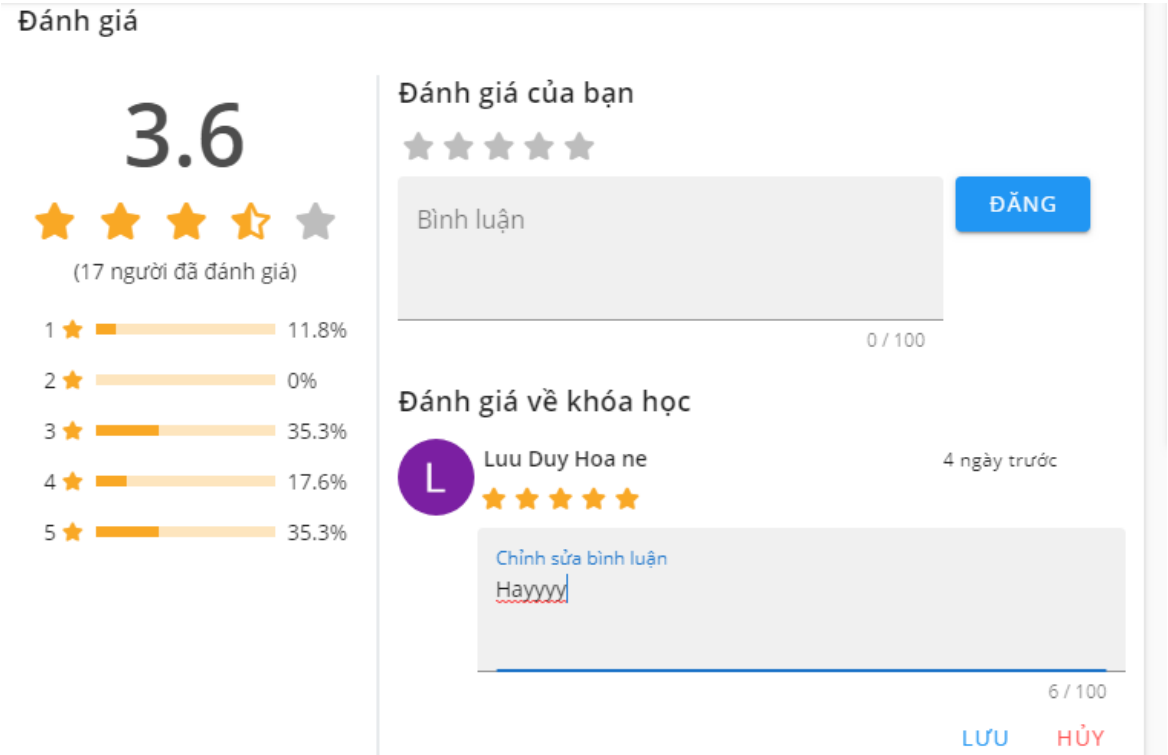


Figure 142: Manage Review

Step	Description
Pre-condition	Login as role Learner. Enrol to course.
1	Click on each course to show detail
2	Hover to your comment, click on "Sửa"
3	After update all field, click on "Lưu"

Table 118: Manage Review desciprtion

2.2.5 Update Profile

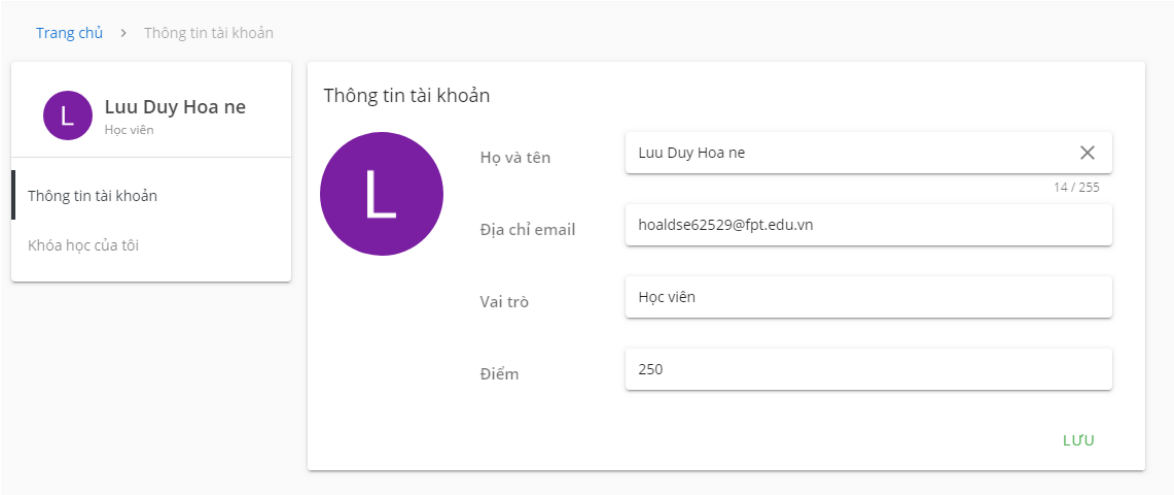


Figure 143: Update profile

Step	Description

Pre-condition	Login as role Learner. Enrol to course.
1	Click on your name at the top right
2	Click on “Thông tin cá nhân”
3	After update all field, click on “Lưu”

Table 119: Update Profile description

2.2.6 Manage Enrolled Courses

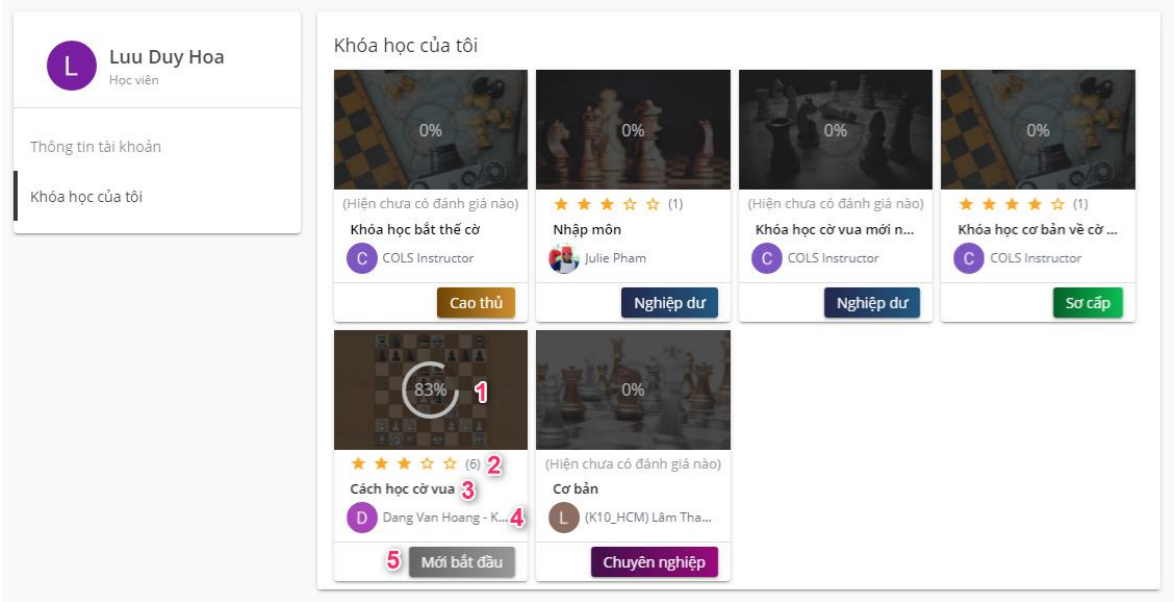


Figure 144: Manage course enrolled

Step	Description
Pre-condition	Login as role Learner. Enrol to course.
1	Click on your name at the top right
2	Click on “Khóa học của tôi”

Table 120: Manage Enrolled Courses description

2.3 Instructor

2.3.1 Create Course

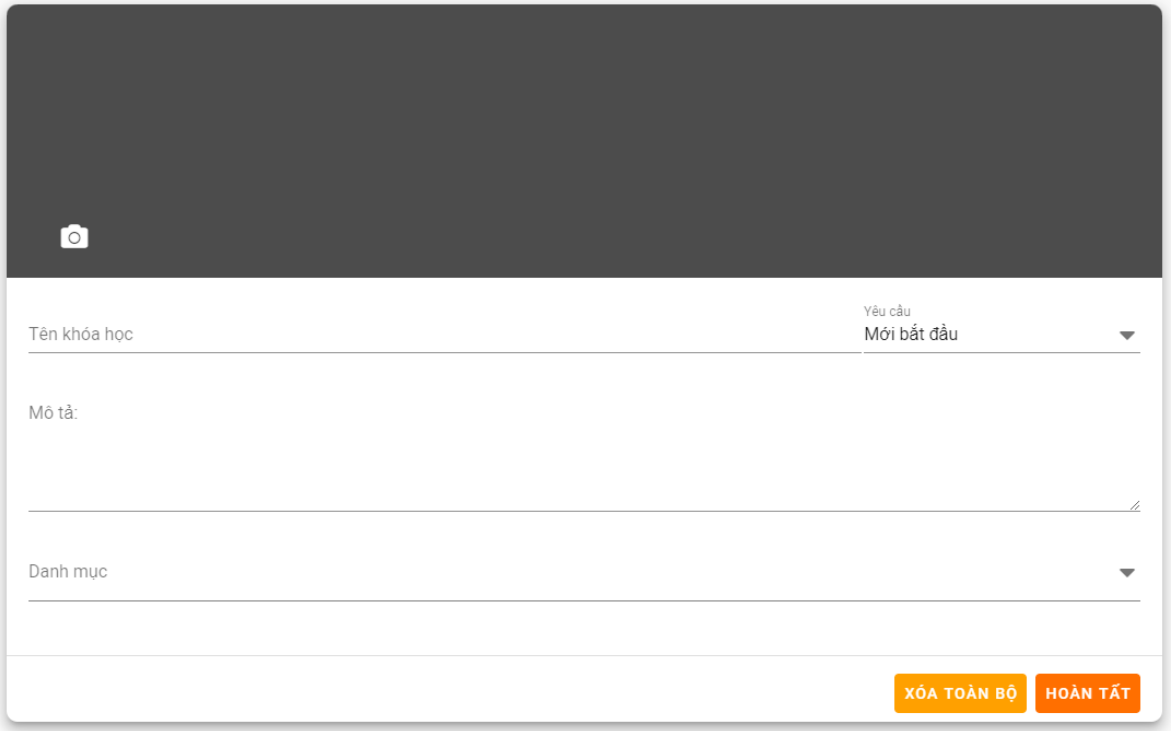
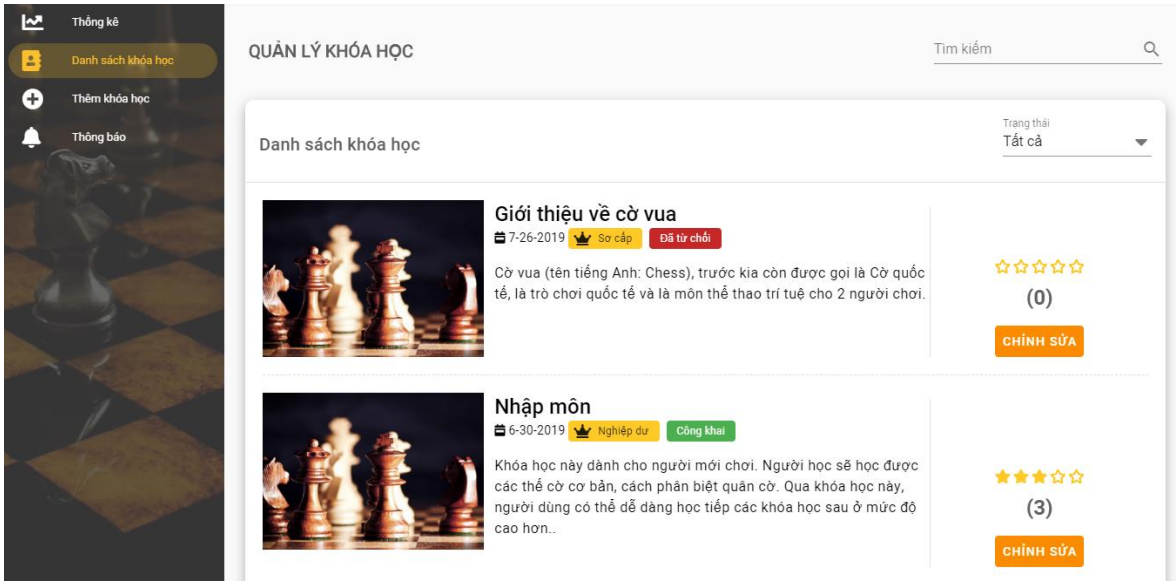


Figure 145: Create course

Step	Description
Pre-condition	Login as role Instructor
1	Click on "Thêm khóa học"
2	After fill in all field, click on "Hoàn tất" to add new course

Table 121: Create Course description

2.3.2 Edit Course Status



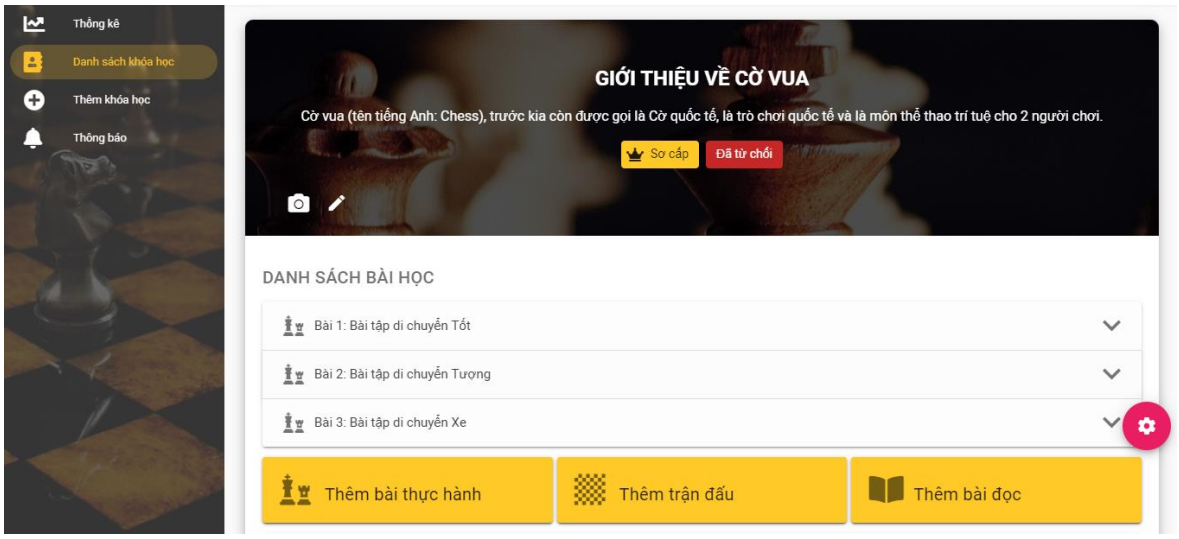


Figure 146: Edit Course Status

Step	Description
Pre-condition	Login as role Instructor
1	Click on “Danh sách khóa học”
2	Click on “Chỉnh sửa” button on each course
3	Click on “Setting” icon to change status course

Table 122: Edit Course Status description

2.2.3 Manage Lesson

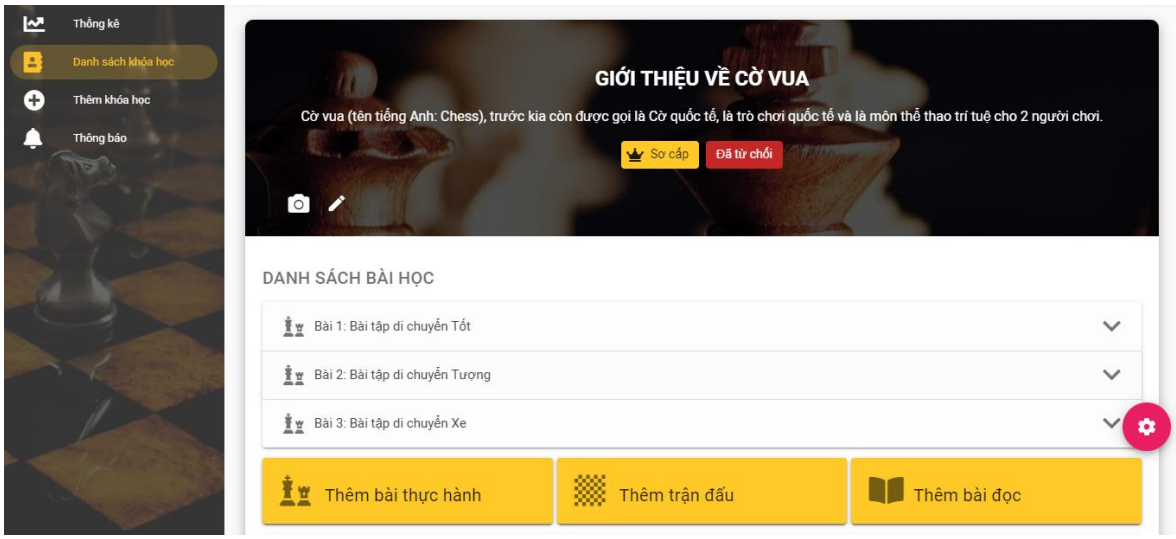


Figure 147: Manage Lesson

Step	Description
Pre-condition	Login as role Instructor
1	Click on “Danh sách khóa học”
2	Click on “Chỉnh sửa” button on each course
3	Click on “View”, “Edit”, “Remove” icon on each lesson to manage

Table 123: Manage Lesson description

2.2.4 Create Interactive Lesson

Figure 148: Create Interactive Lesson

Step	Description
Pre-condition	Login as role Instructor
1	Click on “Danh sách khóa học”

2	Click on “Chỉnh sửa” button on each course
3	Click on “Thêm trận đấu”
4	After fill all field, click “Xem trước” to preview
5	Click on “Lưu” to add new interactive lesson

Table 124: Create Interactive Lesson description

2.2.5 Create Uninteractive Lesson

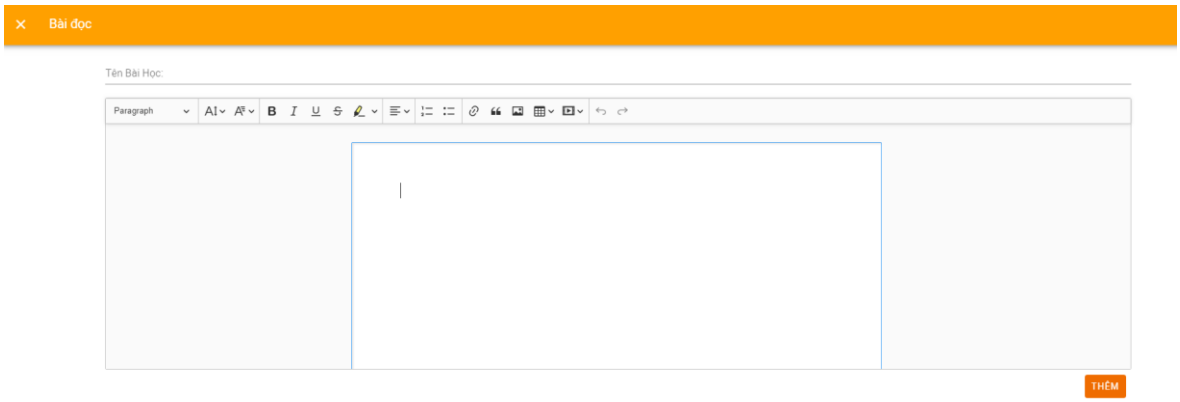


Figure 149: Create Uninteractive Lesson

Step	Description
Pre-condition	Login as role Instructor
1	Click on "Danh sách khóa học"
2	Click on "Chỉnh sửa" button on each course
3	Click on "Thêm bài đọc"
4	After fill content, click "Thêm" to add new lesson

Table 125: Create Uninteractive Lesson description

2.2.6 Create Exercise

Figure 150: Create Exercise

Step	Description
Pre-condition	Login as role Instructor
1	Click on "Danh sách khóa học"
2	Click on "Chỉnh sửa" button on each course

3	Click on “Thêm bài thực hành”
4	After fill content, click “Xem trước” to preview
5	Click on “Lưu” to add new exercise

Table 126: Create Exercise description

2.4 Admin

2.4.1 Active/Deactive Account

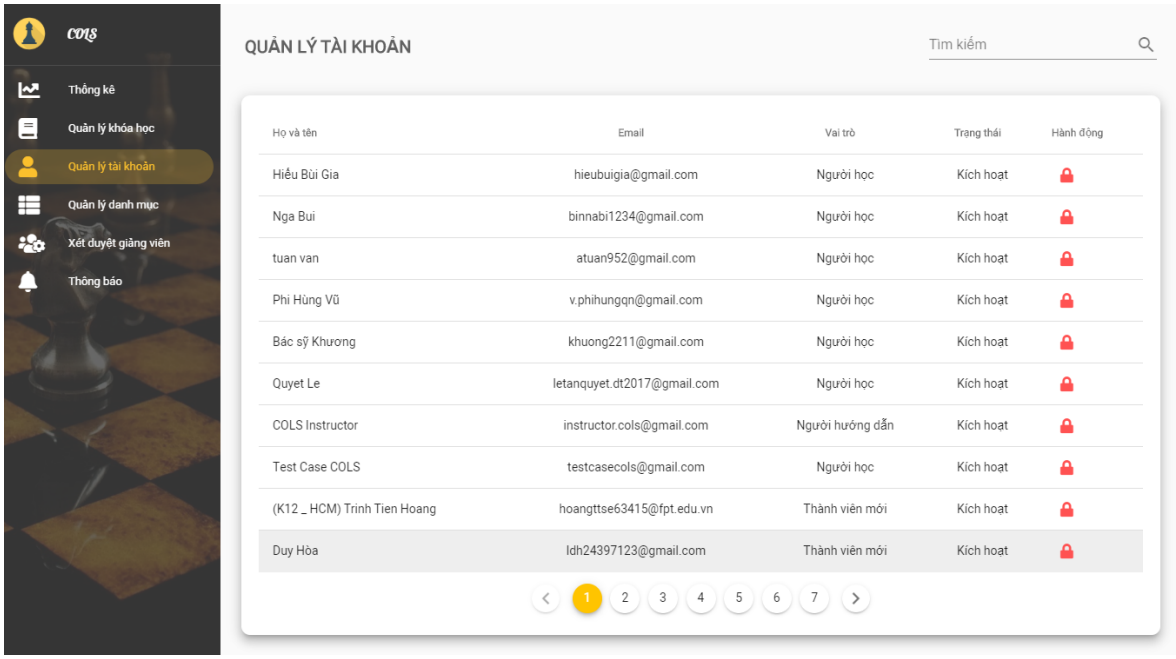


Figure 151: Active/Deactive Account

Step	Description
Pre-condition	Login as role Admin
1	Click on "Quản lý tài khoản"
2	Click on active/deactive icon on the right side

Table 127: Active/Deactive Account description

2.4.2 Search User

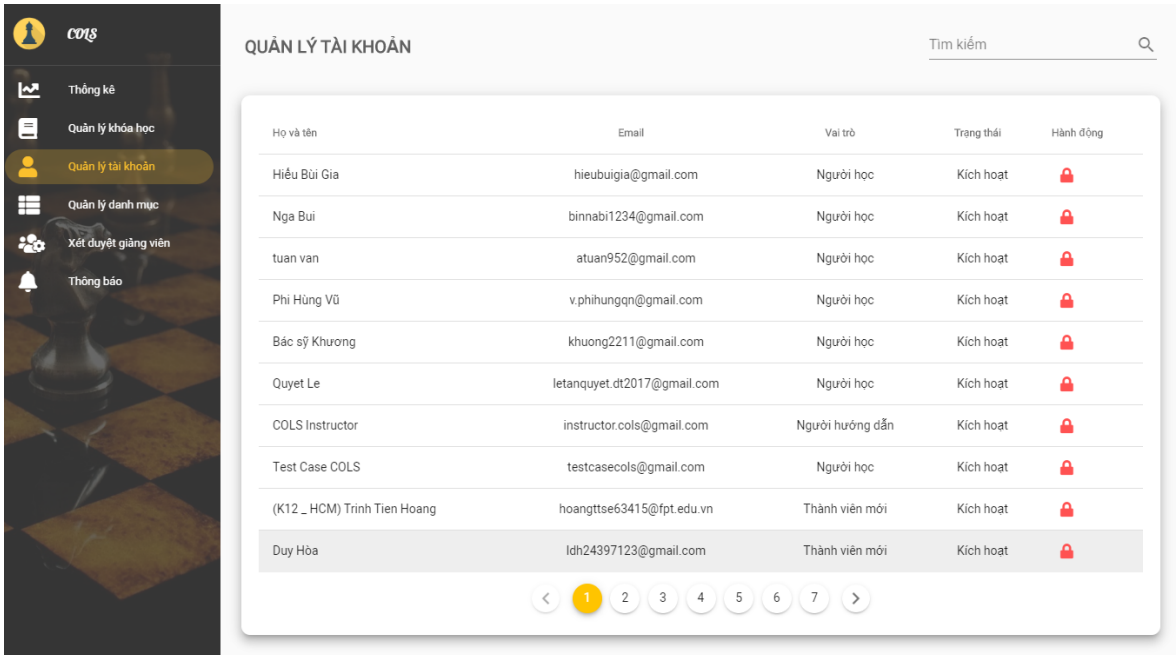


Figure 152: Search User

Step	Description
Pre-condition	Login as role Admin
1	Click on “Quản lý tài khoản”
2	Click on “Tìm kiếm” text field
3	Type any character to search account

Table 128: Search Account description

2.4.3 Accept/Reject Instructor Registration

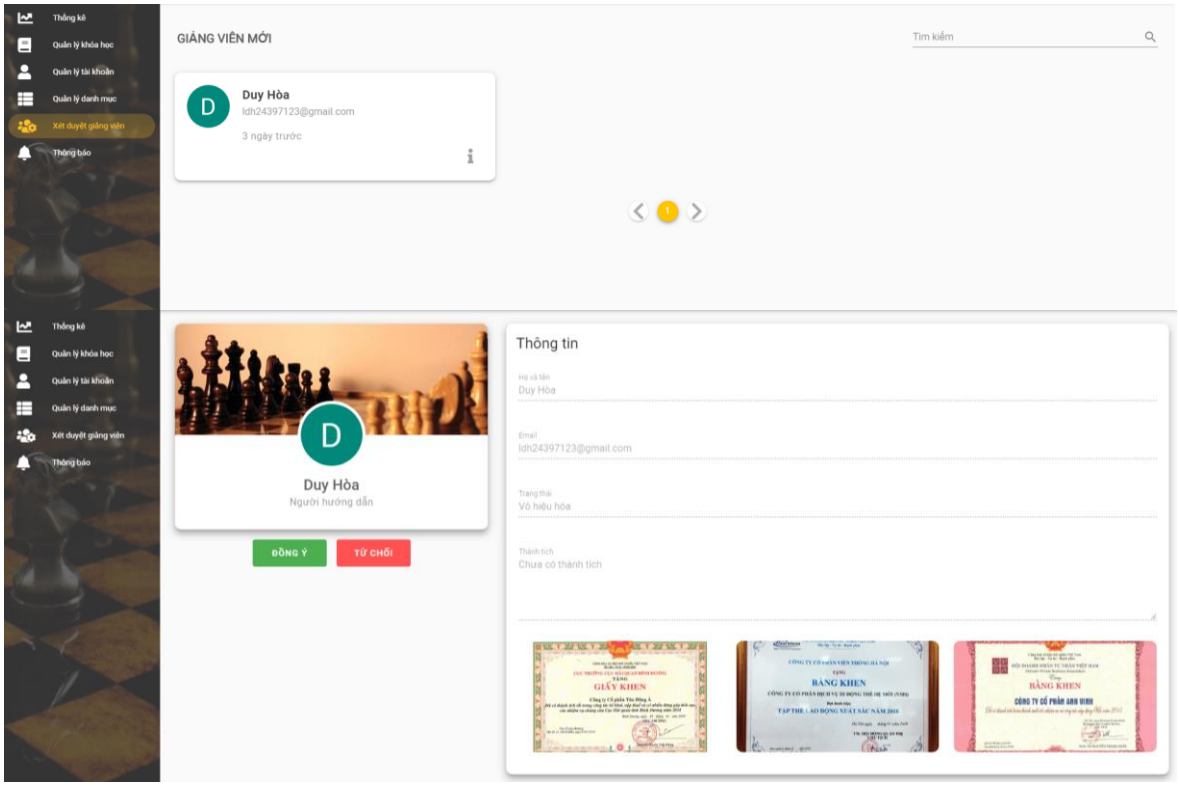


Figure 153: Accept/Reject Instructor Registration

Step	Description
Pre-condition	Login as role Admin
1	Click on “Xét duyệt giảng viên”
2	Click on “Information Icon” on each account to manage
3	Click on “Đồng ý” or “Từ chối” button to Accept or Reject instructor

Table 129: Accept/Reject Instructor Registration description

2.4.4 Manage Category

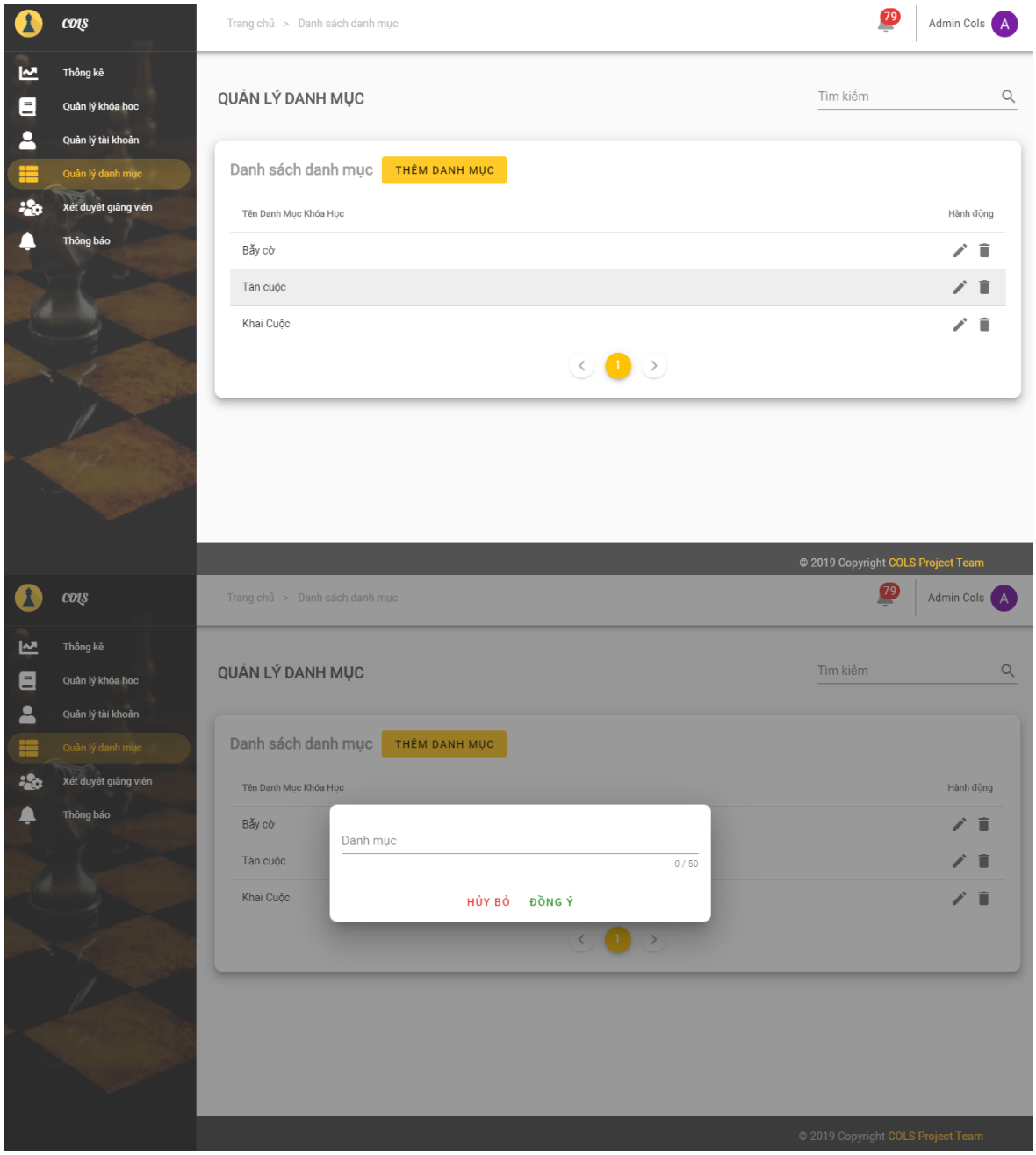


Figure 154: Manage Category

Step	Description
Pre-condition	Login as role Admin
1	Click on “Quản lý danh mục”
2	Click on “Thêm danh mục”
3	After fill in all field, click on “Đồng ý” to add category
4	Click on “Edit” or “Remove” icon on each category to manage

Table 130: Manage Category description

2.4.5 Manage Notification

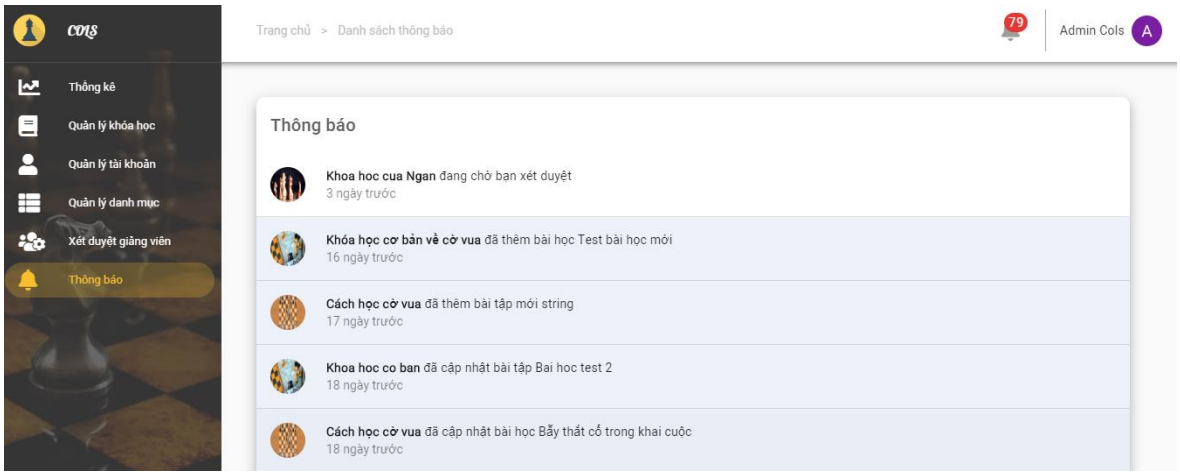


Figure 155: Manage Notification

Step	Description
Pre-condition	Login as role Admin
1	Click on “Thông báo”
2	Click on each notification to see details

Table 131: Manage Notification description

G. Appendix

1. **The Unified Modeling Language** *[Online]* Available: https://www.omg.org/spec/UML/2.0/About-UML/ Accessed on: Sept. 2019.

2. **Multitier Architecture** *[Online]* Available: https://www.ibm.com/support/knowledgecenter/en/SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/covr_3-tier.html Accessed on: Sept. 2019.
3. Chen, P.P., "The Entity-Relationship Model: Toward a Unified View of Data". *ACM Trans. on Database Systems*.

4. **MySQL** *[Online]* Available: https://dev.mysql.com/doc/ Accessed on: Sept. 2019.

5. **Spring Framework** *[Online]* Available: https://spring.io/docs Accessed on: Sept. 2019.

6. **VueJS** *[Online]* Available: https://vuejs.org/v2/guide/ Accessed on: Sept. 2019.

7. **Vuetify** *[Online]* Available: https://vuetifyjs.com/en/getting-started/quick-start Accessed on: Sept. 2019.

8. **Stockfish** *[Online]* Available: https://stockfishchess.org/ Accessed on: Sept. 2019.

9. **Stockfishjs** *[Online]* Available: https://github.com/nmrugg/stockfish.js/ Accessed on: Sept. 2019.

10. **chessjs** *[Online]* Available: https://github.com/jhlywa/chess.js Accessed on: Sept. 2019.

11. **chessground** *[Online]* Available: https://github.com/ornicar/chessground Accessed on: Sept. 2019.

12. **chessboardjs** *[Online]* Available: https://chessboardjs.com/docs Accessed on: Sept. 2019.

13. **FEN String** *[Online]* Available: https://www.chessprogramming.org/Forsyth-Edwards_Notation Accessed on: Sept. 2019.

14. **Chessprogramming wikipedia** *[Online]* Available: https://www.chessprogramming.org/Main_Page Accessed on: Sept. 2019.

15. **Elo Rating System** *[Online]* Available: https://www.fide.com/fide/handbook Accessed on: Sept. 2019.