

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SOFTWARE REQUIREMENTS SPECIFICATION

IT4542E - MANAGEMENT OF SOFTWARE DEVELOPMENT

TOPIC: CORNHUB – ONLINE LEARNING PLATFORM

Group 2

Full Name	Student ID
Nguyen Chi Long	20210553
Ngo Xuan Bach	20215181
Le Xuan Hieu	20215201
Nguyen Chan Hung	20215209
Dinh Viet Quang	20215235
Nguyen Viet Thang	20215245
Nguyen Hoang Viet	20215255
Vu Tuan Hung	20194773

Major: Information Technology

Specialization: Global Information and Communications Technology

Supervisor: PhD. Nguyen Nhat Hai

Signature

Department: Computer Science

School: School of Information and Communications Technology

HANOI, 11/2023

Table of Contents

I. Introduction	4
1. Purpose.....	4
2. Scope.....	4
3. Glossary	4
4. References.....	5
II. Overall description	5
1. Survey	5
2. Overall requirements.....	6
3. Use case decomposition diagram.....	6
3.1. Use case decomposition diagram for “general user”	6
3.2. Use case decomposition diagram for “student”	7
3.3. Use case decomposition diagram for “instructor”	8
3.4. Use case decomposition diagram for “admin”	9
4. Business process	9
4.1. Course learning process as student	10
4.2. Course purchase process as student	10
4.3. Creating courses process as instructor	11
4.4. Managing courses process as instructor.....	12
4.5. Managing courses as administrator.....	13
4.6. Managing user accounts as administrators.....	14
III. Detailed Requirements.....	15
1. General user – guest.....	15
1.1. Specification of use case UC1.1 “Search for course”	15
1.2. Specification of use case UC1.2 “View course information”	16
1.3. Specification of use case UC1.3 “Sign up”	16
1.4. Specification of use case UC1.4 “Log in”	18
1.5. Specification of use case UC1.5 “Forget password”	19
1.6. Specification of use case UC1.6 “Manage profile”	20
1.7. Specification of use case UC1.7 “Log out”	21
2. Student	21
2.1. Specification of use case UC2.1 “View purchased courses”	21
2.2. Specification of use case UC2.2 “Search purchased course”.....	22

2.3. Specification of use case UC2.3 “View course information”	22
2.4. Specification of use case UC2.4 “Review course”	23
2.5. Specification of use case UC2.5 “Manage cart”	24
2.6. Specification of use case UC2.6 “View bill”	25
2.7. Specification of use case UC2.7 “Checkout”	25
2.8. Specification of use case UC2.8 “Do quiz”	26
2.9. Specification of use case UC2.9 “View result”	27
2.10. Specification of use case UC2.10 “Ask questions”	28
2.11. Specification of use case UC2.11 “Learn lectures”	28
2.12. Specification of use case UC2.12 “View notes”	29
2.13. Specification of use case UC2.13 “Write comments”	30
3. Instructor	30
3.1. Specification of use case UC3.1 “Create course”	30
3.2. Specification of use case UC3.2 “Publish course”	31
3.3. Specification of use case UC3.3 “Manage courses”	32
3.4. Specification of use case UC3.4 “Answer QA”	35
3.5. Specification of use case UC3.5 “View Performance”	36
4. Admin.....	37
4.1. Specification of use case UC4.1 “Admin login”	38
4.2 Specification of use case UC4.2 “View requesting list”	39
4.3 Specification of use case UC4.3 “View course detail”	39
4.4. Specification of use case UC4.4 “Verify course”	40
4.5 Specification of use case UC4.5 “View user list”	41
4.6 Specification of use case UC4.6 “View user statistics”	42
4.7 Specification of use case UC4.7 “Delete user account”	42
4.8. Specification of use case UC4.8 “Admin logout”	43
IV. Supplementary specification	44
1. Functionality	44
2. Usability.....	45
3. Reliability.....	45
4. Performance	45

I. Introduction

1. Purpose

The objective of this Software Requirements Specification (SRS) document is to provide a comprehensive description of the requirements for the development of the CornHub learning website. The intended audience for this SRS includes the developers and testers, who will be involved in the development, testing and implementation of the system, following the Agile development model.

The SRS aims to serve as a reference guide and communication tool between the project team and stakeholders, ensuring a clear understanding of the system's objectives, functionalities, and constraints. It will provide the basis for system design, development, testing, and validation activities.

2. Scope

The software product to be produced is the CornHub learning systems, which is a web-based platform for learning systems, a comprehensive and dynamic platform designed to cater to diverse educational needs. The system will support various features and functionalities to provide a seamless user experience.

Functioning as an online learning hub, this website is positioned to offer a wide array of courses, tutorials, and learning materials across various disciplines. From academic subjects to professional skills development, Cornhub Learning aims to provide a rich and accessible learning experience for users of all backgrounds.

Notably, by using the Cornhub Learning website, Users can expect a user-friendly interface, intuitive navigation, and a robust course catalog that spans different expertise levels.

The scope also extends to fostering a collaborative learning environment. This collaborative aspect aims to enhance the overall learning experience, encouraging knowledge-sharing and networking among users.

Furthermore, the website's scope encompasses continuous improvement and expansion. Regular updates to the course offerings, features, and functionalities will ensure that Cornhub Learning remains a dynamic and relevant platform. As technology and educational trends evolve, so will the scope of Cornhub Learning, adapting to meet the changing needs of its users.

In summary, the Cornhub Learning website is intended to be a versatile and user-centric platform that not only provides top-notch educational content but also fosters a sense of community and adaptability akin to successful platforms in the near future.

3. Glossary

Term	Definition
Agile development model	a type of incremental model, developed based on an iterative development process
Use Case Analysis	a technique that aids in modeling the requirements of a software system. A well-crafted Use Case model will describe the system in the most intuitive and easy-to-understand way for all users and clients.

4. References

Introduction to Software Engineering Lecture – PhD. Bui Thi Mai Anh.

Management of Software Development Lecture – PhD. Nguyen Nhat Hai.

Software Project Mangement: For Small to Medium Sized Projects E.Book – John J. Rakos.

Software Development Project Management: Process and Support (Ellis Horwood Books in Information Technology) E.Book – Dina Berkeley, Robert De Hoog, Patrick Humphreys.

II. Overall description

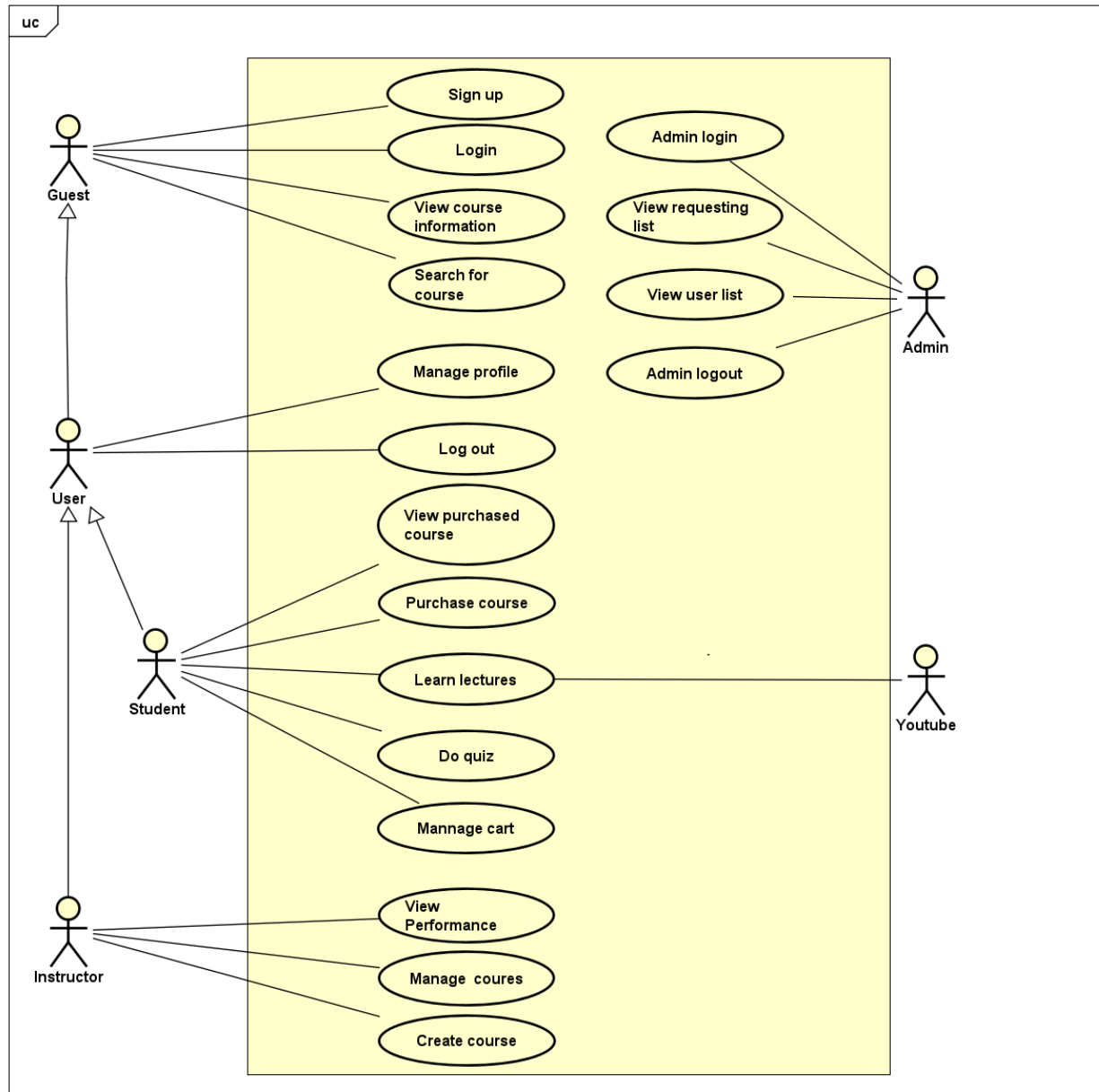
1. Survey

The CornHub System involves three main actors interacting with the system:

- Student: The student represents those who want to participate in learning activities by registering for courses related to diverse domains. They interact with the system to choose a course to follow (this includes doing quizzes and referring course documentations). Students provide necessary personal information, and some extra information that relates to their purpose of registering on the platform, select payment methods and adhere to the terms and conditions.
- Instructor: The instructor actor represents the lecturer presenting the courses. They can create course content for students to follow, provide documentation, open quizzes and grade each student. They also have to provide necessary personal information and adhere to the terms and conditions of service.
- Admin: This actor acts as a role of a manager of the system. They are the only actors who have permission to modify user accounts details and verify created course publish pending requests. They can make contacts with the database through administrator graphical interface.

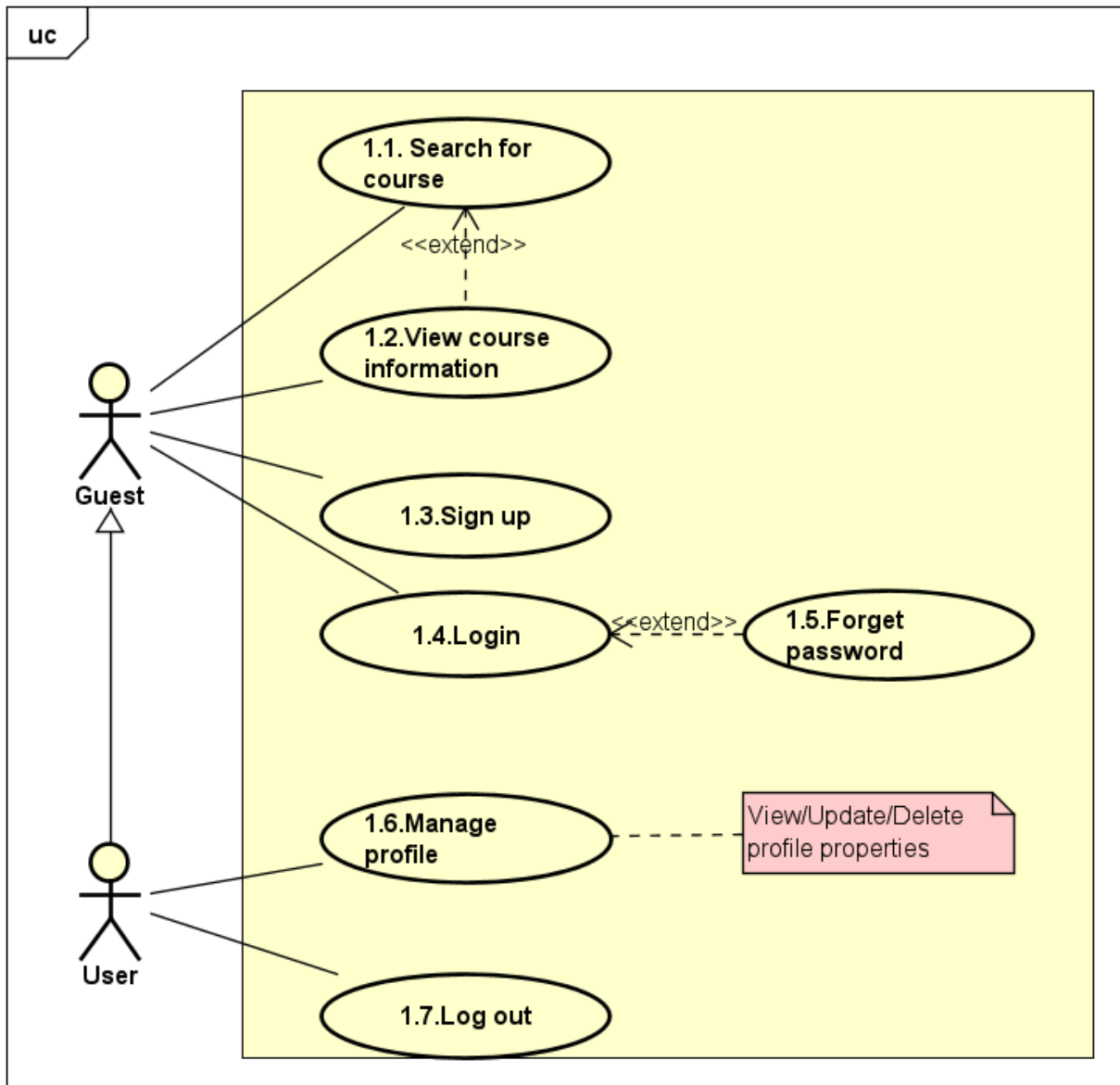
2. Overall requirements

Use case diagram represents the interactions between actors and usecases. It represents the functional requirements of the system, showing the interaction between external and internal actors with the system.

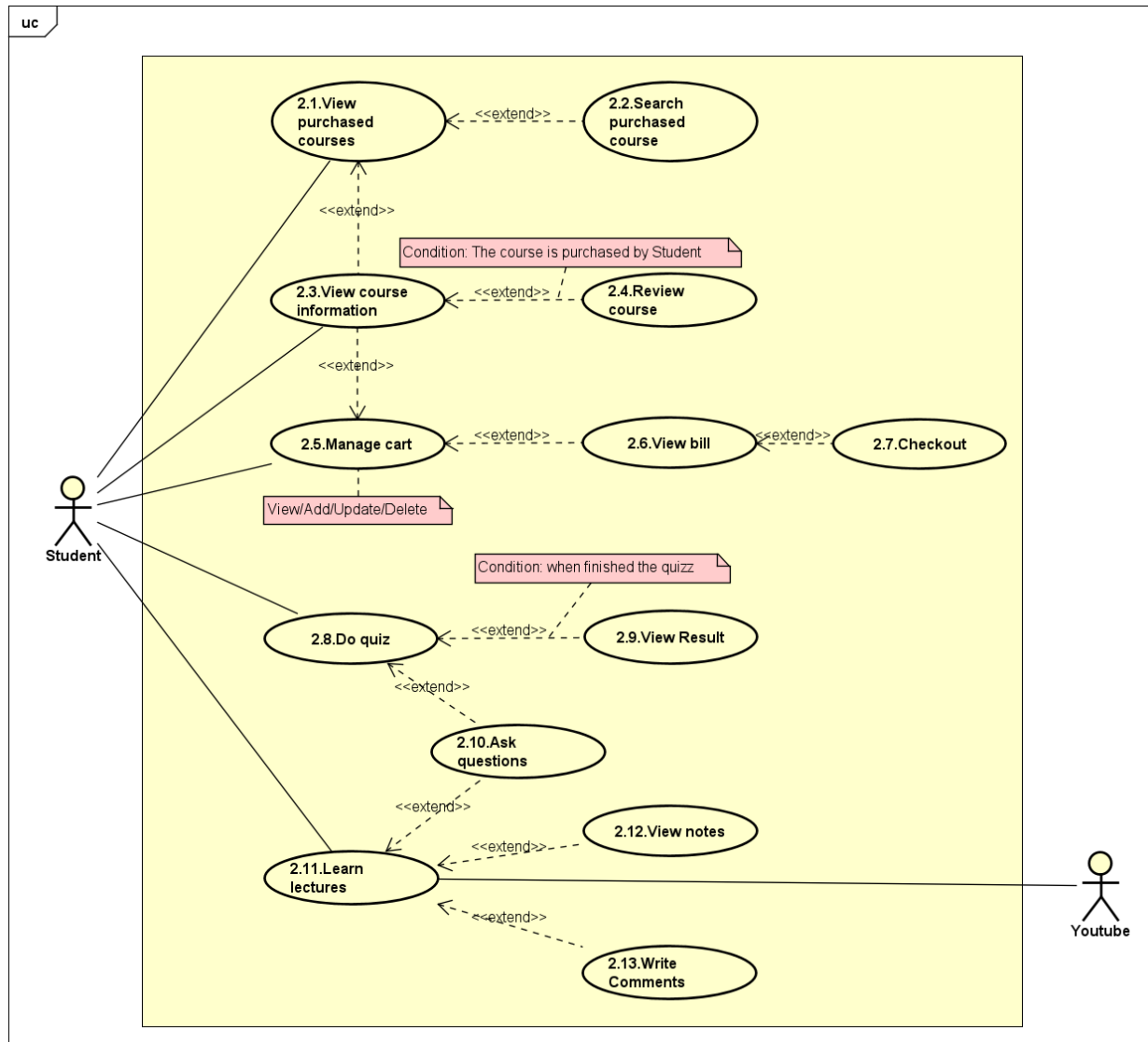


3. Use case decomposition diagram

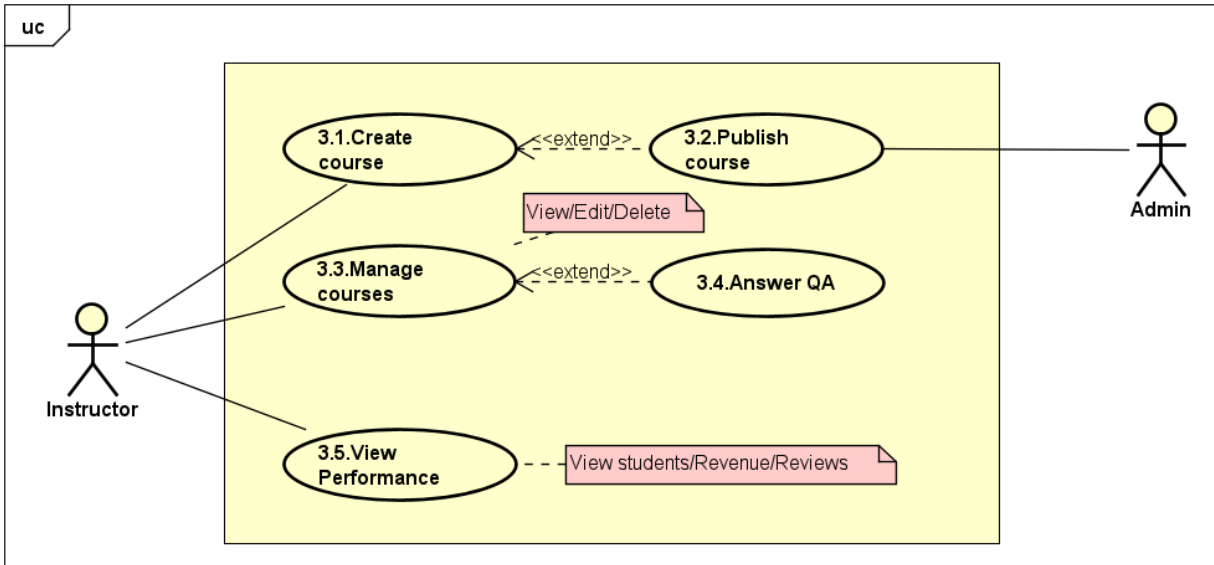
3.1. Use case decomposition diagram for “general user”



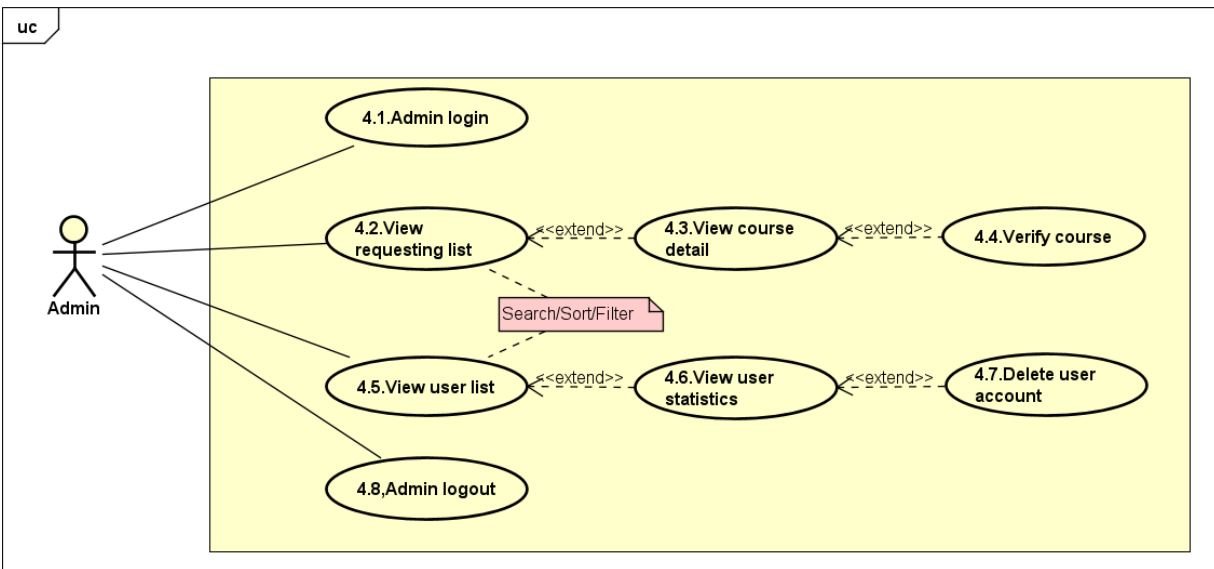
3.2. Use case decomposition diagram for “student”



3.3. Use case decomposition diagram for “instructor”



3.4. Use case decomposition diagram for “admin”



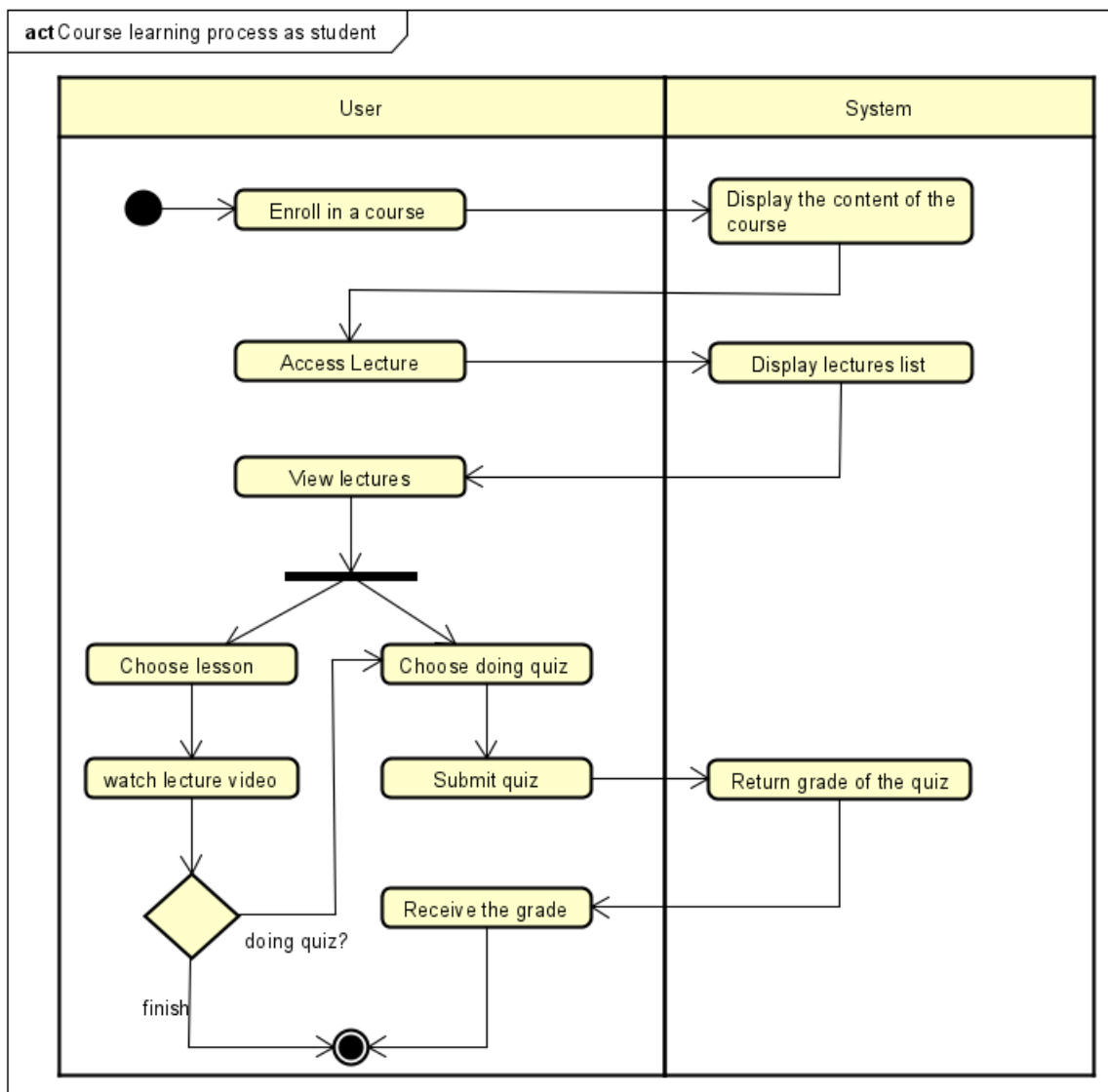
4. Business process

In this subsystem, several main business processes are identified. For students, we have chosen to analyze the following processes: the course browsing process, the learning process, and the course purchase process. As for instructors, the focus is on the processes of creating courses and managing courses. With regards to admins, the spotlight is on course and user account management, which involves course content censoring, account verification and authentication.

Details of the actions in these processes are visualized with activity diagrams in the sub-sections of each process.

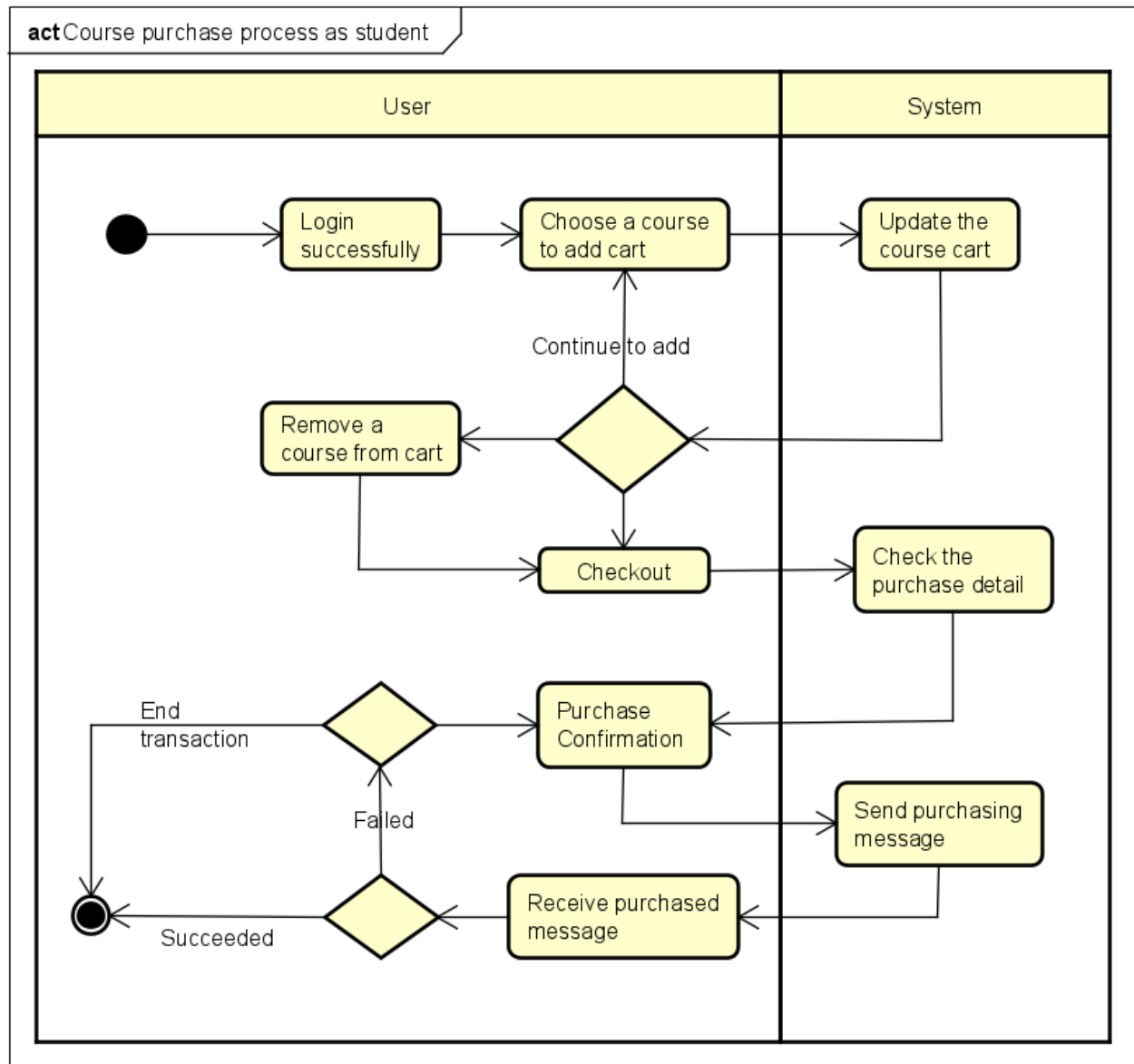
4.1. Course learning process as student

When students join the learning process, a menu of course materials appear. The student goes through the steps of enrolling in the course, accessing lectures, and viewing materials to complete a lesson. After completing several lessons in the course, the student can choose to take fun quizzes for a better understanding of the lecture contents and then receive point for each quiz they do from the auto-grade system.



4.2. Course purchase process as student

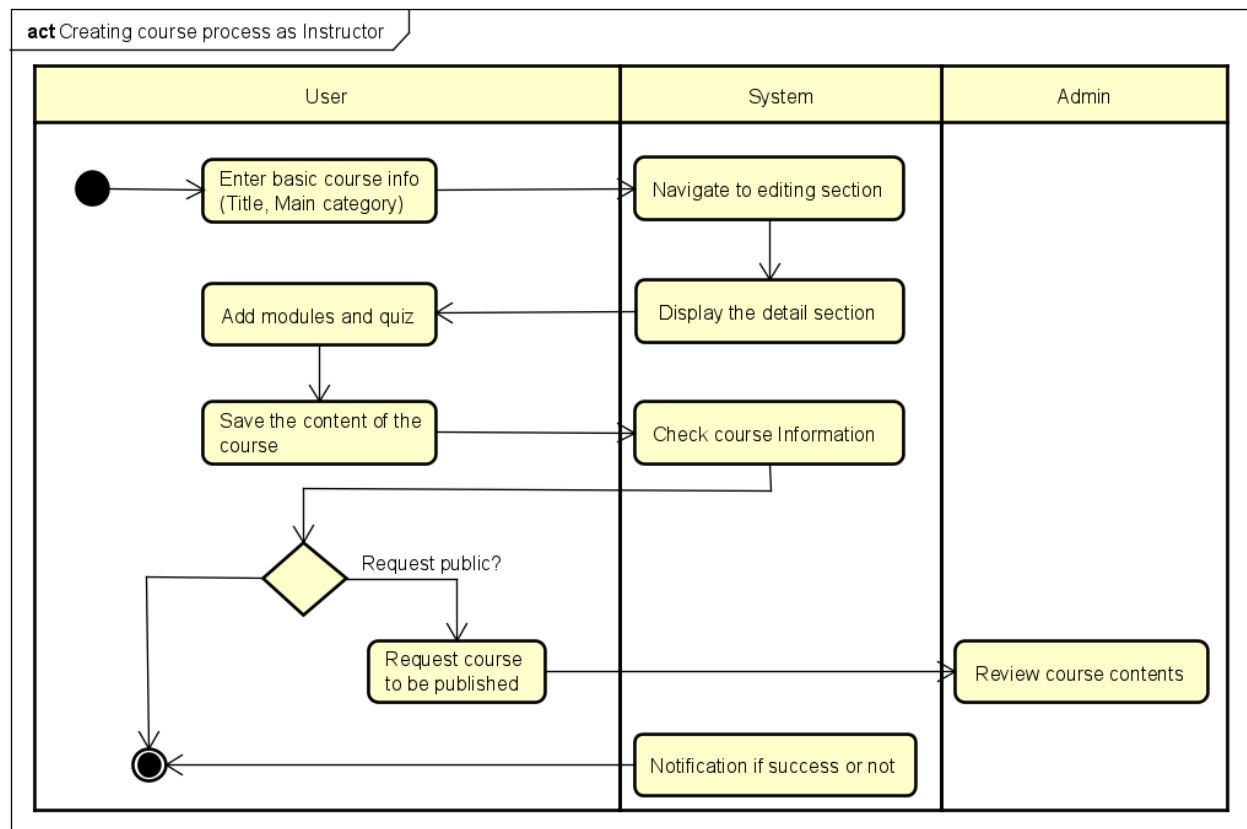
The process of course purchase begins with the student logging in successfully. They then choose a course to add to their cart and have the option to continue adding more courses or proceed to checkout. The system updates the course cart accordingly. If desired, the student can remove a course from their cart before checking out. Upon checkout, the system checks the purchase details and confirms the purchase if everything is in order. A purchasing message is then sent by the system, and upon receipt of this message, the transaction is considered successful.



4.3. Creating courses process as instructor

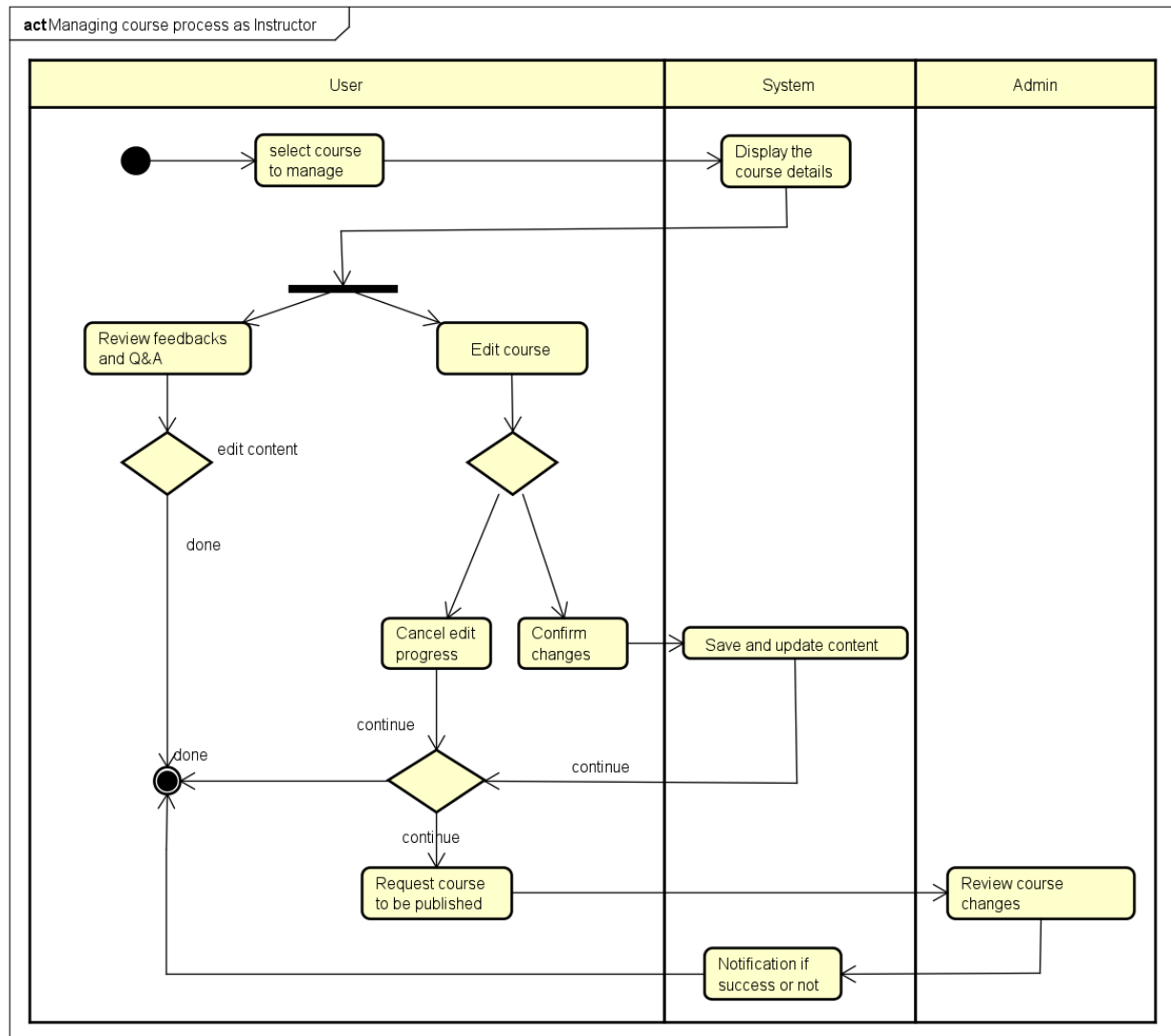
This activity diagram outlines the business process of an instructor creating a course. It involves interactions between the user (instructor), system, and admin. The instructor enters basic course information, adds modules and quizzes, and saves the content. They can then request for the

course to be public. The system displays details and checks course information while the admin reviews the contents and notifies if it is successful or not.



4.4. Managing courses process as instructor

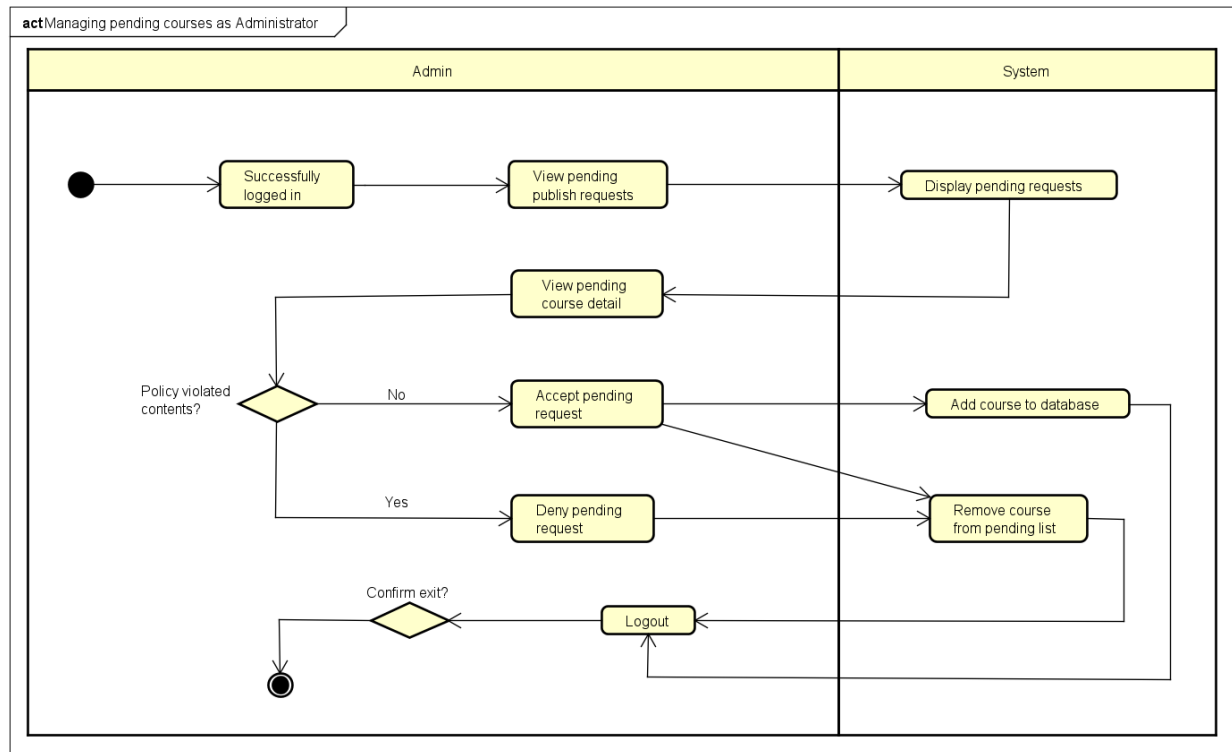
The business process of an instructor's course management involves several steps and decisions. First of all, the process begins with the instructor selecting a course to manage. The instructor can then review feedback and Q&A and edit content if necessary. If content is edited, there's an option to cancel the edit progress or confirm changes. After confirming changes or if no edits are made, the instructor can request the course to be published. The system displays the course details and checks the course information, then saves and updates the content of the course. The admin reviews the course changes and notifies the instructor if the course is successfully verified or not.



4.5. Managing courses as administrator

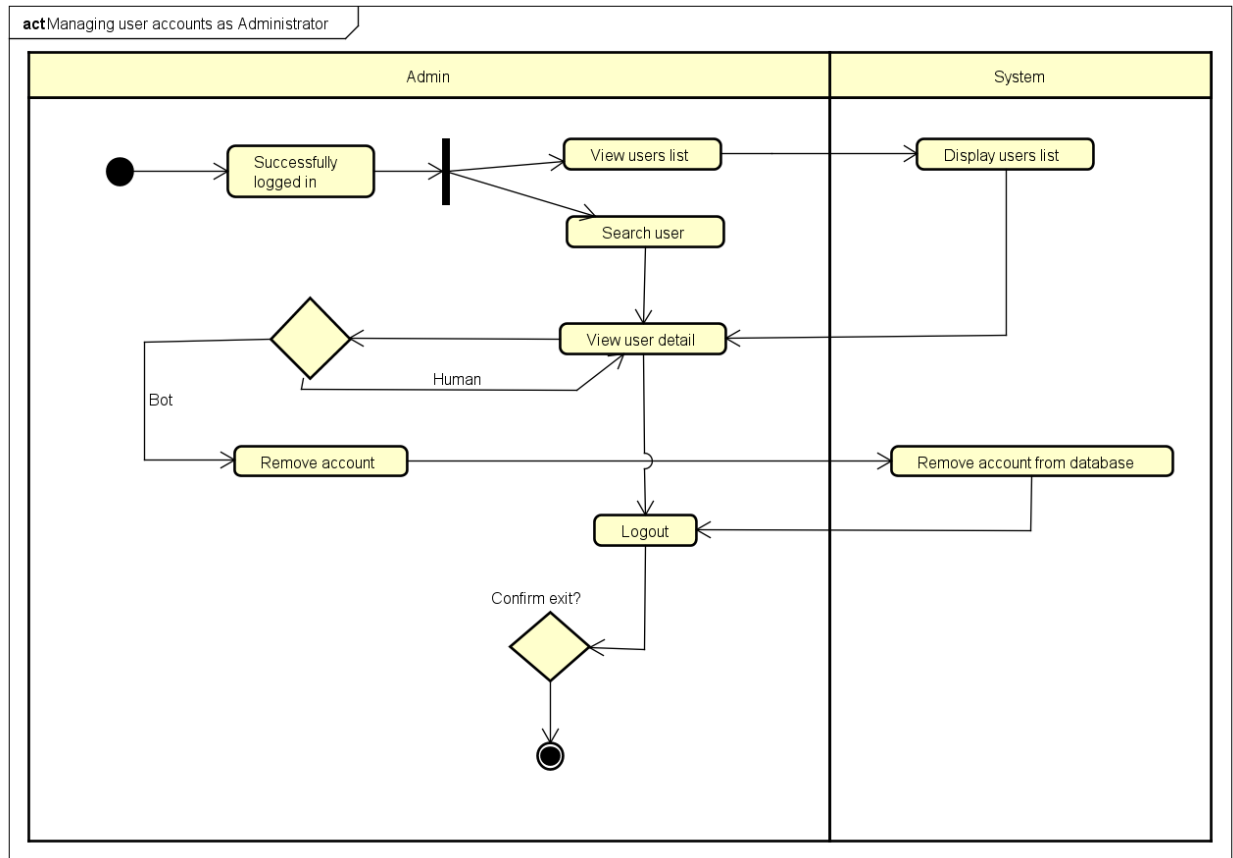
Admins should follow the steps specified in the following process diagram to do management on courses.

The process involves searching for courses, viewing the courses list, and checking pending published requests for instructor-created but unpublished courses. For each branch of the process, the system will operate differently based on the administrators' decisions. The system may make changes to the database when admins accept a pending request or when a course violating community standards or policies is detected.



4.6. Managing user accounts as administrators

This process involves viewing user accounts list and searching for accounts. Both options result in viewing user details. Admins can track the activities of an account to determine whether the account is used by human or robot, then they can make decisions on whether to remove or retain the account. The system may also make a removal on the database if admin conceives an account to be spam or robot manipulated.



III. Detailed Requirements

Details of the use cases given in part 2 are described in the sections below.

1. General user – guest

1.1. Specification of use case UC1.1 “Search for course”

Use case ID	UC1.1	Name of use case	Search for course
Actor	Guest		
Trigger	The guest wants to search for course		
Pre-condition	None		
Basic flow (Complete)	Ordinal number	Done by	Action
	1	Guest	Enter keywords on the search bar
	2	Guest	Hit the “search” button

	3	System	Display matching courses
Alternative flow			
	Ordinal number	Done by	Action
	3a	System	Show notification that no courses match the query
Post-condition	The guest views matching course list on the screen		

1.2. Specification of use case UC1.2 “View course information”

Use case ID	UC1.2	Name of use case	View course information
Actor	Guest		
Trigger	The guest wants to see a course in the course list		
Pre-condition	None		
Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Guest	Click a course in the course list
	2	System	Show information about the course
Alternative flow	None		
Post-condition	The guest views the course information on the screen		

1.3. Specification of use case UC1.3 “Sign up”

Use case ID	UC1.3	Name of use case	Sign up
Actor	Guest		
Trigger	The guest wants to register an account of the learning website		

Pre-condition	None		
Basic flow (Complete)	Ordinal number	Done by	Action
	1	Guest	Choose sign up feature
	2	System	Show the interface of signup page
	3	Guest	Complete the sign-up form
	4	Guest	Send the sign-up form
	5	System	Verify information in the form
	6	System	Accept the account registration and allow the guest to interact as a user
	7	System	Redirect to the home page
Alternative flow	Ordinal number	Done by	Action
	7a	System	Fail to create new account and notify the error (missing fields, invalid information, email already created, weak password...)
	8a	Guest	Enter the forms again or go to another page
Post-condition	Account is created and the guest views the home page with the role of a user		

* Input data (for signup):

Ordinal Number	Attribute	Description	Must have?	Eligible Condition	Example
1.	Email		Yes	Valid email address	h.anh@gmail.com
2.	Password		Yes	At least 8 characters, including special characters and uppercase for first letter	ToiLa12#\$
3.	FirstName		Yes		John
4.	LastName		Yes		Doe

1.4. Specification of use case UC1.4 “Log in”

Use case ID	UC1.4	Name of use case	Log in
Actor	Guest		
Trigger	The guest wants to access to the webpage		
Pre-condition	None		
Basic flow (Complete)	Ordinal number	Done by	Action
	1	Guest	Choose log in feature
	2	System	Show the interface of login page
	3	Guest	Complete the log in form
	4	Guest	Send the log in form
	5	System	Verify information in the form
	6	System	Accept the account access and allow the guest to interact as a user
	7	System	Redirect to the home page
Alternative flow	Ordinal number	Done by	Action
	7a	System	Fail to log in and notify the error (missing fields, invalid information, email already created, weak password...)
	8a	Guest	Enter the forms again or go to another page
Post-condition	The guest views the home page with the role of a user		

* Input data (for login):

Ordinal Number	Attribute	Description	Must have?	Eligible Condition	Example
1.	Email		Yes	Valid email address	h.anh@gmail.com

2.	Password		Yes	At least 8 characters, including special characters and uppercase for first letter	ToiLa12#\$
----	----------	--	-----	--	------------

1.5. Specification of use case UC1.5 “Forget password”

Use case ID	UC1.5	Name of use case		Forget password
Actor	Guest			
Trigger	The guest wants to reset the password			
Pre-condition	None			
Basic flow (Complete)				
	Ordinal number	Done by	Action	
	1	Guest	Choose forget password feature	
	2	System	Show the interface of forget password page	
	3	Guest	Complete the form	
	4	Guest	Send the form	
	5	System	Verify information in the form	
	6	System	Accept the password retrieval and allow the guest to interact as a user	
7	System	Redirect to the log in page		
Alternative flow				
	Ordinal number	Done by	Action	
	7a	System	Fail to retrieve password and notify the error (missing fields, invalid information, weak password...)	
8a	Guest	Enter the forms again or go to another page		
Post-condition	Account is created with email provided and the guest views the home page with the role of a user			

* Input data (for forget password):

Ordinal Number	Attribute	Description	Must have?	Eligible Condition	Example
1.	Email		Yes	Valid email address	h.anh@gmail.com
2.	Password		Yes	At least 8 characters, including special characters and uppercase for first letter	ToiLa12#\$

1.6. Specification of use case UC1.6 “Manage profile”

Use case ID	UC1.6	Name of use case	Manage profile
Actor	User		
Trigger	The user wants to take a look at the profile information		
Pre-condition	The user has already logged in		
Basic flow (Complete)	Ordinal number	Done by	Action
	1	Guest	Choose view profile feature
	2	System	Show the interface of view profile page
	3	Guest	Have several options such as View, Update, Delete profile properties
	4	System	Update the profile page corresponding to the user’s requests and changes.
	5	Guest	Return to the login page
	Alternative flow	None	
Post-condition	The user is in log in status		

1.7. Specification of use case UC1.7 “Log out”

Use case ID	UC1.7	Name of use case		Log out
Actor	User			
Trigger	The user logs out of the website			
Pre-condition	The user has already logged in			
Basic flow (Complete)				
	Ordinal number	Done by	Action	
	1	Guest	Choose log out feature	
	2	System	Execute logging out feature	
	3	Guest	Return to the homepage version for guest	
Alternative flow	None			
Post-condition	The user are in log in status			

2. Student

2.1. Specification of use case UC2.1 “View purchased courses”

Use case ID	UC2.1	Name of use case	View purchased courses
Actor	Student		
Trigger	The student wants to check the course that already bought by the student		
Pre-condition	The student has already logged in		
Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Student	Choose “Learning” on the nav bar

	2	System	Show the purchased course list
Alternative flow	Ordinal number	Done by	Action
	2a	System	Show blank list and notify that the student has not purchased any courses
	2b	System	Redirect to the login page if the guest has not logged in
Post-condition	The student views his/her purchased course list on the screen		

2.2. Specification of use case UC2.2 “Search purchased course”

Use case ID	UC2.2	Name of use case	Search purchased course
Actor	Student		
Trigger	The student searches for a purchased course		
Pre-condition	The student is in the purchased course list page		
Basic flow (Complete)	Ordinal number	Done by	Action
	1	Student	Type keywords on the search bar
	2	Student	Hit the “search” button
	3	System	Display a list of matching purchased courses
Alternative flow	Ordinal number	Done by	Action
	3a	System	Show blank list and notify that no purchased courses match the query
Post-condition	The student views a list of matching purchased course list on the screen		

2.3. Specification of use case UC2.3 “View course information”

Use case ID	UC2.3	Name of use case	View course information
--------------------	-------	-------------------------	-------------------------

Actor	Student											
Trigger	The student takes a look at a course in the purchased course list or in the course list in the home page											
Pre-condition	None											
Basic flow (Complete)	<table><tr><th>Ordinal number</th><th>Done by</th><th>Action</th></tr><tr><td>1</td><td>Student</td><td>Click a course in the purchased course list or in the course list in the home page</td></tr><tr><td>2</td><td>System</td><td>Show the course information and the “Add to cart” button if the student has not enrolled in</td></tr></table>			Ordinal number	Done by	Action	1	Student	Click a course in the purchased course list or in the course list in the home page	2	System	Show the course information and the “Add to cart” button if the student has not enrolled in
Ordinal number	Done by	Action										
1	Student	Click a course in the purchased course list or in the course list in the home page										
2	System	Show the course information and the “Add to cart” button if the student has not enrolled in										
Alternative flow	<table><tr><th>Ordinal number</th><th>Done by</th><th>Action</th></tr><tr><td>2a</td><td>System</td><td>Show the course information and the “Continue your learning” button if the student has already enrolled in</td></tr></table>			Ordinal number	Done by	Action	2a	System	Show the course information and the “Continue your learning” button if the student has already enrolled in			
Ordinal number	Done by	Action										
2a	System	Show the course information and the “Continue your learning” button if the student has already enrolled in										
Post-condition	The student views the course information and “Add to cart” / “Continue your learning” button based on the state of his/her learning											

2.4. Specification of use case UC2.4 “Review course”

Use case ID	UC2.4	Name of use case	Review course
Actor	Student		
Trigger	The student reviews in the review box and chooses rating level in the course information page		
Pre-condition	The student has enrolled in the course		
Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Student	Type review in the review box and choose rating level in the course information page

	2	Student	Hit the button to send the review
	3	System	Record the review and notify that the review is successfully sent
Alternative flow	None		
Post-condition	The review is recorded by the system and is shown on the review list of the course		

2.5. Specification of use case UC2.5 “Manage cart”

Use case ID	UC2.5	Name of use case	Manage cart
Actor	Student		
Trigger	The student wants to check the course to buy in cart page		
Pre-condition	The student has already logged in		
Basic flow (Complete)	Ordinal number	Done by	Action
	1	Student	Click “Cart” icon on the navigation bar
	2	System	Show the cart page interface and courses that the student added to the cart
	3	Student	Go to the home page, choose a course that he/she has not enrolled in and click “Add to cart” button
	4	System	Update the cart with the new course
Alternative flow	Ordinal number	Done by	Action
	3a	Student	Go to the home page, choose a course that he/she has already added to the cart and click “Add to cart” button
	4a	System	Notify the error that the course is already in the cart and cannot be added
	3b	Student	Click “Remove” button of a course

	4b	System	Remove the course out of the cart
Post-condition	The student views the updated cart		

2.6. Specification of use case UC2.6 “View bill”

Use case ID	UC2.6	Name of use case	View bill
Actor	Student		
Trigger	The student confirms the courses to be bought		
Pre-condition	The cart is not empty		
Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Student	Click “Checkout” in the cart page
	2	System	Show the bill with price and confirmation choices
Alternative flow	Ordinal number	Done by	Action
	2a	System	Notify the error that the cart is empty
Post-condition	The student views the bill		

2.7. Specification of use case UC2.7 “Checkout”

Use case ID	UC2.7	Name of use case	Checkout
Actor	Student		
Trigger	The student clicks “Confirm” in the confirmation box in the bill		
Pre-condition	The student is viewing the bill		
Basic flow (Complete)			

	Ordinal number	Done by	Action
	1	Student	Click “Confirm” in the confirmation box in the bill
	2	System	Redirect to the home page and notify that the student has successfully enrolled in the chosen courses (Assume all courses now are free)
Alternative flow	None		
Post-condition	Purchased course list is updated		

2.8. Specification of use case UC2.8 “Do quiz”

Use case ID	UC2.8	Name of use case	Do quiz
Actor	Student		
Trigger	The student wants to do a quiz exercise in the course that he/she is learning		
Pre-condition	None		
Basic flow (Complete)	Ordinal number	Done by	Action
	1	Student	Click a quiz exercise in the course that he/she is learning
	2	System	Display questions in the quiz
	3	Student	Answer questions
	4	Student	Hit the “Submit” button
	5	System	Show the number of correct answers. There are 3 options: “View result”, “Next lesson”, “Try again”
	6	Student	Click “Next lesson” and the system shows the content of the next lesson

Alternative flow	Ordinal number	Done by	Action
	5a	System	Notify the error of failing to submit the quiz (missing answers, wrong format answers...)
	6a	Student	Fix the errors and resubmit again. Use case comes back to step 4
	5b	Student	Choose “Try again”. Use case comes back to step 2
	5c	Student	Choose “View result”. Use case continues as UC2.9
Post-condition	The system records the submitted quiz and the student views the number of correct answers		

2.9. Specification of use case UC2.9 “View result”

Use case ID	UC2.9	Name of use case	View result
Actor	Student		
Trigger	The student wants to views result after finishing a quiz		
Pre-condition	The student finishes a quiz		
Basic flow (Complete)	Ordinal number	Done by	Action
	1	Student	Click “View result” option after finishing a quiz
	2	System	Display the detailed result and explanation for each answer
	3	Student	Hit the “Finish reviewing” button
	4	System	Return to the previous box which show the number of correct answers
Alternative flow	None		
Post-condition	None		

2.10. Specification of use case UC2.10 “Ask questions”

Use case ID	UC2.10	Name of use case	Ask questions
Actor	Student		
Trigger	The student enters their questions for lectures or quiz questions in the reviewing form		
Pre-condition	They are learning a lecture or reviewing a quiz		
Basic flow (Complete)	Ordinal number	Done by	Action
	1	Student	Enter their questions for lectures or quiz questions in the reviewing form
	2	Student	Hit the “Send” button
	3	System	Record the question. Lecture questions are shown and public to all users, while quiz questions can only be seen and answered by instructors.
Alternative flow	None		
Post-condition	The system records the question. Lectures and quiz questions are shown to a certain type of users.		

2.11. Specification of use case UC2.11 “Learn lectures”

Use case ID	UC2.11	Name of use case	Learn lectures
Actor	Student		
Trigger	The student wants to continue the learning of the lectures		
Pre-condition	The student has enrolled in the course		
Basic flow (Complete)	Ordinal number	Done by	Action

	1	Student	Click “Continue your learning” button in a course information page
	2	System	Show the content of the course and current progress of the student
	3	Student	Choose a section and a lesson that he/she wants to learn
	4	System	Update the progress and display the lesson. Use YouTube API if the lesson is of video type
	5	Student	Choose “Next lesson” or any lessons that he/she wants to continue. Use case comes back to step 4.
Alternative flow	None		
Post-condition	The system records and updates the learning progress		

2.12. Specification of use case UC2.12 “View notes”

Use case ID	UC2.12	Name of use case	View notes
Actor	Student		
Trigger	The student notes something from in a lesson		
Pre-condition	The student is learning a lesson		
Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Student	Click “My note” button in a lesson
	2	System	Show the student’s note
	3	Student	Add/Update/Remove the note and hit “Save” button
Alternative flow	4	System	Save the updated note
	Ordinal number	Done by	Action

	4a	System	Notify the error that note is too long to be saved
Post-condition	The student views the updated note		

2.13. Specification of use case UC2.13 “Write comments”

Use case ID	UC2.13	Name of use case	Write comments
Actor	Student		
Trigger	The student wants to comment for a question in the lecture		
Pre-condition	The student is learning a lesson		
Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Student	Types comment for a question in the lecture
	2	System	Record the comment and display it in the Q&A section
Alternative flow	None		
Post-condition	Record the comment and display it in the Q&A section		

3. Instructor

3.1. Specification of use case UC3.1 “Create course”

Use case ID	UC3.1	Name of use case	Create course
Actor	Instructor		
Trigger	The instructor creates a course		
Pre-condition	The instructor is in the instructor’s page		

Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Instructor	Enter the title and category of the course
	2	Instructor	Enter the “Create” button
Alternative flow	3	System	Create the course
	Ordinal number	Done by	Action
	3a	System	Notify the error
a: Instructor does not respond to system’s prompt			
Post-condition	The course is created whose instructor is the creator		

* Input data when creating a course:

Ordinal number	Attribute	Description	Must have?	Eligible conditions	Example
1	Title		Yes	Length in [3,100]	Management of Software Development
2	Category		Yes	No	Information Technology

3.2. Specification of use case UC3.2 “Publish course”

Use case ID	UC3.2	Name of use case	Publish course
Actor	Admin (primary), Instructor (secondary)		
Trigger	The instructor wants to publish a course to every learner by making it publicity, visible in homepage		
Pre-condition	The admin is in course managements page		
Basic flow			

(Complete)	Ordinal number	Done by	Action
	1	Admin	Accept or decline the request to publish a course
Alternative flow	Ordinal number	Done by	Action
	1a	System	Error: Cannot find the course (Removed by the creator before)
	a: Database does not respond to system's prompt		
Post-condition	The course is published		

3.3. Specification of use case UC3.3 “Manage courses”

Use case ID	UC3.3	Name of use case	Manage courses
Actor	Instructor (primary), System (secondary)		
Trigger	The instructor manages course moving into a created course		
Pre-condition	The instructor is in the instructor's page		
Basic flow (Complete)	Course details:		
	Ordinal number	Done by	Action
	1	Instructor	Change the Title, Subtitle, Description, Language, Level, Category, Subcategory field
	2	Instructor	Enter the “Save” button
	3	System	Save new information about the course
	Curriculum:		
	Ordinal number	Done by	Action
	1	Instructor	Enter the “Add new chapter” button

	2	Instructor	Add information about the course content
	3	Instructor	Enter the “Create” button
	4	System	Add the new content into the course’ curriculum
	5	Instructor	Enter the “Save” button
	Intended learners:		
	Ordinal number	Done by	Action
	1	Instructor	Change the “What will students learn in your course?”, “What are the requirements or prerequisites for taking your course?”, “Who is this course for?” field
	2	Instructor	Enter the “Save” button
	3	System	Save new information about the course
	Pricing:		
	Ordinal number	Done by	Action
	1	Instructor	Choose “Free” or “Paid”
	2	Instructor	Enter the price for 1 month and 1 year
	3	Instructor	Enter the “Save” button
	4	System	Save new information about the course
Alternative flow	Course details:		
	Ordinal number	Done by	Action
	3a	System	Notify the error
	Curriculum:		
	Ordinal number	Done by	Action
	4a	System	Notify the error
	Intended learners:		

	Ordinal number	Done by	Action
	3a	System	Notify the error
	Intended learners:		
	Ordinal number	Done by	Action
	2b	Instructor	None
	4a	System	Notify the error
	a: The instructor does not respond to system's prompt		
	b: If the instructor chooses the "Free" option		
Post-condition	The course's information is changed		

* Input data (Course details):

Ordinal number	Attribute	Description	Must have?	Eligible conditions	Example
1	Title		Yes	Length in [3,100]	Management of Software Development
2	Subtitle				MSD
3	Description				This course teaches you the introduction to manage the software's development process
4	Language				English
5	Level				Undergraduate
6	Category		Yes		Information Technology
7	Subcategory				Management

* Input data (Curriculum):

Ordinal number	Attribute	Description	Must have?	Eligible conditions	Example
----------------	-----------	-------------	------------	---------------------	---------

1	Name		Yes		Lecture 1: Introduction
2	Type		Yes	From a given list: Video, Text, File, Assignment, Q&A	Video
3	Content		Yes		

* Input data (Intended learners):

Ordinal number	Attribute	Description	Must have?	Eligible conditions	Example
1	What will students learn in your course?		Yes	At least 4	Learn how to run a script
2	What are the requirements or prerequisites for taking your course?				Undergraduate
3	Who is this course for?				Beginner Python developers

* Input data (Pricing):

Ordinal number	Attribute	Description	Must have?	Eligible conditions	Example
1	Price per month		Yes	Higher than 0	\$12
2	Price per year		Yes	Higher than 0 and price per month	\$120

3.4. Specification of use case UC3.4 “Answer QA”

Use case ID	UC3.4	Name of use case	Answer QA
Actor	Instructor (primary), System (secondary)		
Trigger	The instructor enters a Q&A content of a course		
Pre-condition	The instructor is in a Communications page		
Basic flow			

(Complete)	Ordinal number	Done by	Action
	1	Instructor	Choose a question to answer
	2	Instructor	Enter the answer into the field for comment
	3	Instructor	Enter the “Send” button
	4	System	Save the answer
Alternative flow			
	Ordinal number	Done by	Action
	4a	System	Notify the error
	a: Instructor does not respond to system prompt, or the question is deleted by the student		
Post-condition	The question is answered by the instructor		

* Input data when answering:

Ordinal number	Attribute	Description	Must have?	Eligible conditions	Example
1	Answer		Yes		Yes, this is true because of Pythagoras’s theorem

3.5. Specification of use case UC3.5 “View Performance”

Use case ID	UC3.5	Name of use case	View performance
Actor	System (primary), Instructor (secondary)		
Trigger	The instructor views the performance (popularity, revenue) of a course		
Pre-condition	None		
Basic flow (Complete)	Overview:		
	Ordinal number	Done by	Action

	1	System	Show the total revenue, enrollments and instructor rating of all courses
	Reviews:		
	Ordinal number	Done by	Action
	1	System	Show the review of all courses
Alternative flow	Review:		
	Ordinal number	Done by	Action
	1a	System	Notify that there's no review
	a: No one review any course		
Post-condition	The instructor can see the performance		

* Output data (Overview):

Ordinal number	Attribute	Description	Display format	Example
1	Total revenue		Chart	
2	Total enrollments		Chart, Integer	
3	Instructor rating		Real number	5.0

* Output data (Review):

Ordinal number	Attribute	Description	Display format	Example
1	Course	Name, instructor of the course	Text	Management of Software Development – Nguyen Nhat Hai
2	Review		Text	Good course for new developers

4. Admin

4.1. Specification of use case UC4.1 “Admin login”

Use case ID	UC4.1	Name of use case	Admin login
Actor	Guest		
Trigger	The manager accesses into the admin page		
Pre-condition	None		
Basic flow (Complete)	Ordinal number	Done by	Action
	1	Guest	Choose login feature
	2	System	Show the interface of login page
	3	Guest	Complete the login form
	4	Guest	Send the login form
	5	System	Verify information in the form
	6	System	Accept the account access and allow the guest to interact as an admin
	7	System	Redirect to the home page
Alternative flow	Ordinal number	Done by	Action
	6a	System	Fail to login and notify the error (missing fields, invalid information, email already created, weak password,...)
	6b	System	Account cannot be accessed as admin. Login as user
	7a	Guest	Enter the forms again or go to another page
Post-condition	The guest views the home page with the role of an admin		

* Input data (for signup):

Ordinal Number	Attribute	Description	Must have?	Eligible Condition	Example
1.	Email		Yes	Valid email address	h.anh@gmail.com

2.	Password		Yes	At least 8 characters, including special characters and uppercase for first letter	ToiLa12#\$
----	----------	--	-----	--	------------

4.2 Specification of use case UC4.2 “View requesting list”

Use case ID	UC4.2	Name of use case	View requesting list
Actor	Admin		
Pre-condition	The admin is in Admin page		
Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Admin	Choose View requesting list feature
	2	System	Get requesting list and show in the interface
Alternative flow			
	Ordinal number	Done by	Action
	2a.	System	Requesting list is empty, return empty
Post-condition	Admin can see requesting course list		

4.3 Specification of use case UC4.3 “View course detail”

Use case ID	UC4.3	Name of use case	View course detail
Actor	Admin		
Pre-condition	Admin is on course-showing interface		
Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Admin	Choose View course detail feature
	2	System	Get course detail and show in the interface
Post-condition	Admin can see course detail		

* Output data (Review):

Ordinal number	Attribute	Description	Display format	Example
1	Course	Name, instructor of the course	Text	Management of Software Development – Nguyen Nhat Hai
2	Topic		Text	Computer Science

4.4. Specification of use case UC4.4 “Verify course”

Use case ID	UC4.4	Name of use case	Verify course
Actor	Admin		
Pre-condition	Admin is viewing requesting course list and viewing course’s details		
Basic flow			

(Complete)	Ordinal number	Done by	Action
	1	Admin	Admin review course contents
	2	Admin	If the course is not violated, choose to accept
	3	System	Add course to the official database
	4	System	Delete course from pending list
Alternative flow	Ordinal number	Done by	Action
	2a.	Admin	If the course has content violation, choose deny request
	3a.	System	Ask the admin to confirm Denying request
	4a.	Admin	Confirm
	5a	System	Delete course from pending list
Post-condition	Admin has verified a pending course		

4.5 Specification of use case UC4.5 “View user list”

Use case ID	UC4.5	Name of use case	View user list
Actor	Admin		
Pre-condition	The admin is in Admin page		
Basic flow (Complete)	Ordinal number	Done by	Action
	1	Admin	Choose View user list feature
	2	System	Get user list and show in the interface with option to view user detail

Post-condition	Admin can see user list
-----------------------	-------------------------

4.6 Specification of use case UC4.6 “View user statistics”

Use case ID	UC4.6	Name of use case	View user statistics
Actor	Admin		
Pre-condition	The admin is viewing user		
Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Admin	Choose view user statics
	2	System	Get user statics and show in the form of bar/chart (e.g.: how many users are actively studying, % instructor/student)
Post-condition	Admin can see performance, statistics about user		

4.7 Specification of use case UC4.7 “Delete user account”

Use case ID	UC4.7	Name of use case	Delete user account
Actor	Admin		
Pre-condition	The admin is viewing user list/detail		
Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Admin	Review whether the account is bot or human

	2	Admin	If account is bot, choose to delete account
	3	System	Ask the admin to confirm delete the account
	4	Admin	Confirm delete the account
	5	System	Delete the account from database
Alternative flow			
	Ordinal number	Done by	Action
	2a.	Admin	The account is human user, choose not to delete
Post-condition	None		

4.8. Specification of use case UC4.8 “Admin logout”

Use case ID	UC4.8	Use case name	Admin logout
Actor	Admin		
Pre-condition	Guest logged in as admin		
Basic flow (Complete)			
	Ordinal number	Done by	Action
	1	Admin	Choose logout feature
	2	System	Ask the admin to confirm logout, notify unsaved work
	3	Admin	Confirm logging out
	4	System	Sign out of account, back to home interface
Alternative flow			
	Ordinal number	Done by	Action
	3a	Admin	Choose not to log out

Post-condition	Admin signed out of account
-----------------------	-----------------------------

IV. Supplementary specification

- Performance:
 - Fast loading times to improve user experience.
 - Handle multiple users simultaneously without sacrificing quality of service.
- Reliability:
 - Periodic data backup and restore.
 - Error reporting system and routine maintenance.
- Ease of Use:
 - User-friendly and easy-to-use interface.
- Ease of maintenance:
 - Automated monitoring and reporting system.
 - Easy to update and expand the system without affecting the service.
- Databases:
 - Use a high-performance and scalable database: MongoDB.
- Frontend/Backend Technology:
 - Use popular and stable technology such as Node.js, React.js.
- Confidentiality:
 - Integrated data encryption and network traffic security.
 - Manage access and control user data.
- System Integration:
 - Support integration with third-party services such as payment systems, payment gateways, and social technologies.

1. Functionality

- In the sequence of events of use cases, all steps that operate with the database, if there is an error during the connection or operation, there should be a corresponding error message so that the agent knows that the error is related to the database, not related to user error.
- For use cases utilized by the Administrator and User, the Guest is required to log in with the corresponding role.
- The general display format is as follows:
 - Right-aligned number

- Left-aligned text
- Font: Arial 14, Black color
- White background

2. Usability

Functions should be designed to be easy to operate. Specific instructions for user errors should be provided, enabling users to identify errors, understand their implications, and know how to fix them.

3. Reliability

The system can operate for 200 hours continuously without errors and restore to normal within 2 hours after the error occurs.

4. Performance

The multi-platform system works 24/7. The maximum system response time is 1 second at normal or 2 seconds at peak.