Software Requirements Specification

for

FLS – FPT Learning System

Version 1.0.0

Prepared by

Student name: Tran Quang Huy
Student Id: GCD18457

Instructor: *Hoang Nhu Vinh*
Course: **Application Development**
Teaching Assistant: *Hoang Nhu Vinh*
Date: **26/08/2020**

Contents

CONTENTS **2**

REVISIONS **2**

1 INTRODUCTION **3**

1.1 DOCUMENT PURPOSE 3
1.2 PRODUCT SCOPE 3
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW 3
1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS 3
REFERENCES AND ACKNOWLEDGMENTS 4

2 OVERALL DESCRIPTION **5**

2.1 PRODUCT OVERVIEW 5
PRODUCT FUNCTIONALITY 7
2.2 DESIGN AND IMPLEMENTATION CONSTRAINTS 8

3 SPECIFIC REQUIREMENTS **12**

3.1 USER INTERFACES 12
3.2 FUNCTIONAL REQUIREMENTS 47
3.3 USE CASE MODEL 86

4 OTHER NON-FUNCTIONAL REQUIREMENTS **94**

4.1 PERFORMANCE REQUIREMENTS 94
4.2 SAFETY AND SECURITY REQUIREMENTS 94
4.3 SOFTWARE QUALITY ATTRIBUTES 94
4.4 LOCALIZATION 94

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0.0	Tran Quang Huy	FPT Learning System	26/08/20

1 Introduction

1.1 Document Purpose

This project showcases the web-based system, which manages the activity of "Training" for the internal training program of FPT Co.
The project aims to build a system that can manage trainee accounts, manage trainers, manage course categories, manage courses, manage topics, assign the topic to course, assign a trainer to a topic, assign trainee to the course used by the HR department.
There are 4 main roles: **administrator**, **training staff**, **trainer**, and **trainee** with different functions.

1.2 Product Scope

FPT Learning System is a system to manage staff, trainer, trainee, and manage courses.
The system targets a website that can work well on the Desktop and will develop on mobile in the future.

1.3 Intended Audience and Document Overview

This report is working on the Application Development subject. Reports will be presented to the teacher in charge of the subject, Mr. Hoang Nhu Vinh and classmates.
The main focus of the report is the deployment functionality, the use case model

1.4 Definitions, Acronyms, and Abbreviations

No.	Abbreviations/Terms	Explanation
1	SRS	Software Requirement Specification
2	MVC	Model-View-Controller
3	ERD	Entity Relationship Diagram
4	DFD	Data Flow Diagram
5	CRUD	Create-Read-Update-Delete
6	UC	Use Case
7	FLS	FPT Learning System

Table 1. Definition, Acronyms and Abbreviations table

References and Acknowledgments

allpcworld. (n.d.). *allpcworld*. Retrieved from allpcworld: https://allpcworld.com/download-bootstrap-studio-2-2-4-professional-free/

fptshop. (n.d.). *fptshop*. Retrieved from fptshop: https://fptshop.com.vn/phan-mem/windows-10-pro-32bit64bit-fpp

microsoft. (n.d.). *microsoft*. Retrieved from microsoft: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/generics/constraints-on-type-parameters

techtalk. (n.d.). *techtalk*. Retrieved from techtalk: https://techtalk.vn/ro-ri-phien-ban-visual-studio-2019-version-16-0-0-preview-1.html

w3. (n.d.). *w3*. Retrieved from w3: https://www.w3.org/MarkUp/HTMLConstraints.html

webdesign. (n.d.). *webdesign*. Retrieved from webdesign: https://www.webdesign-essentials.ch/apps/balsamiq

wikipedia. (n.d.). *wikipedia*. Retrieved from wikipedia: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio

wikipedia. (n.d.). *wikipedia*. Retrieved from wikipedia: https://en.wikipedia.org/wiki/Google_Chrome

2 Overall Description

2.1 Product Overview

System workflow:

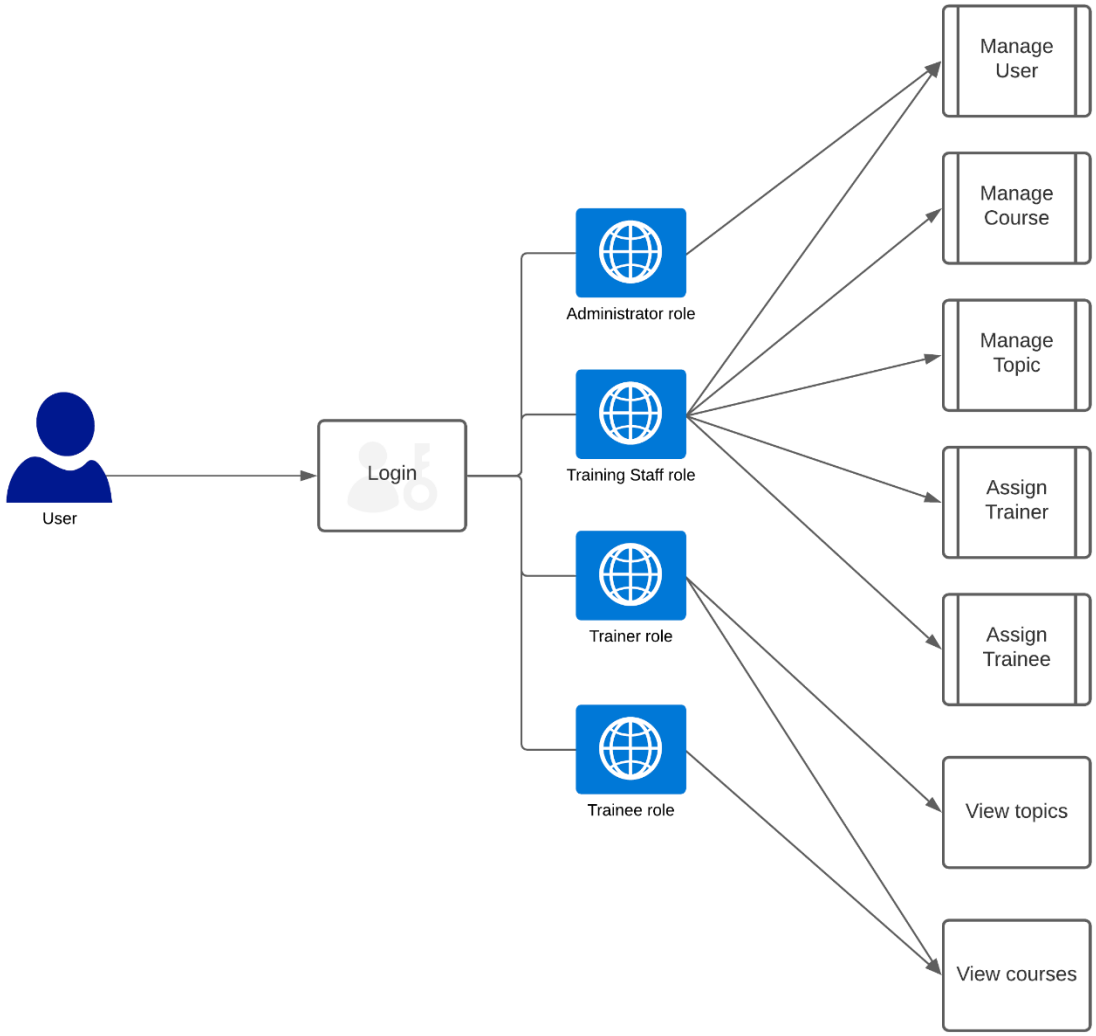


Figure 1. System workflow

System MVC model:

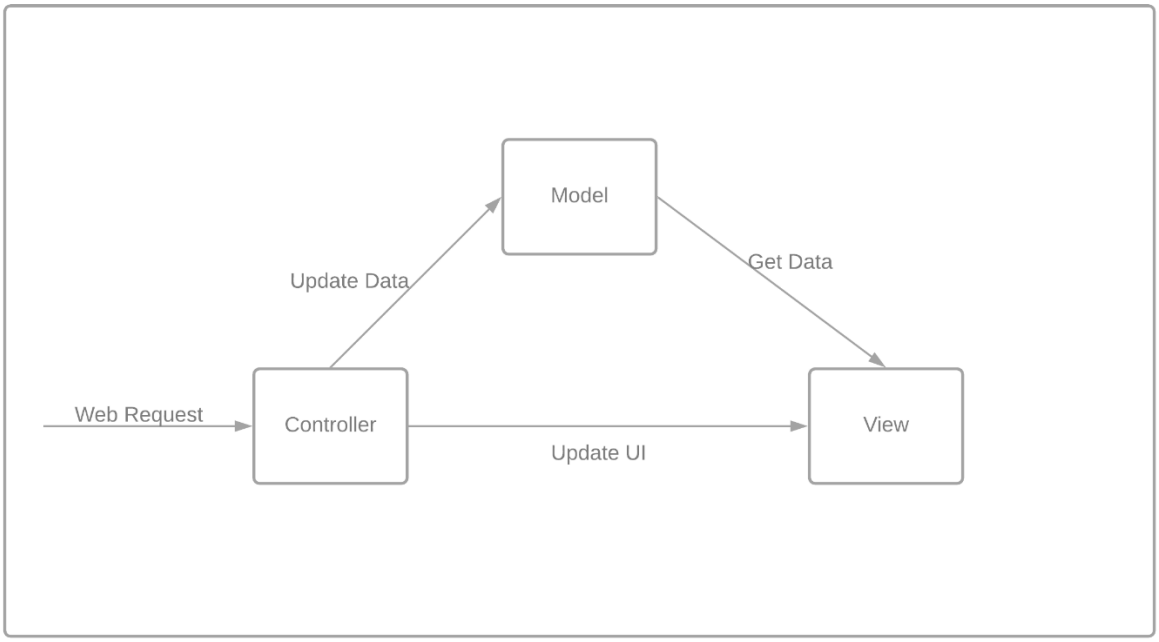


Figure 2. System MVC model

Product Functionality

- As an administrator's role:
  - An administrator can log in to FLS
  - An administrator can manage trainer and training staff such as creating, editing, and deleting.
  - An administrator can assign or change username and password for user
- As a training staff's role:
  - Training staff can log in to FLS
  - Training staff can create trainee account by entering details such as trainee name, trainee accounts, age, date of birth.
  - Training staff can view the list of trainees
  - Training staff can look for trainee by trainee account, email.
  - Training staff can manage trainee account such as updating, deleting.
  - Training staff can manage categories such as searching, adding, updating, and deleting.
  - Training staff can manage courses such as searching, adding, updating, and deleting.
  - Training staff can manage topics such as searching, adding, updating, and deleting.
  - Training staff can manage trainer information such as adding, updating, and deleting information.
  - Training staff can assign a trainer to the topic.
  - Training staff can assign trainee to course
- As a trainer's role:
  - A trainer can log in to FLS
  - A trainer can manage his profile by updating the Trainer's name, email, password.
  - A trainer can view courses that trainer has been assigned to a topic.
  - A trainer can view the topic that the trainer has been assigned to the topic.
- As a trainee's role:
  - A trainee can log in to FLS
  - A trainee can view courses that trainee has been assigned to course.

2.2 Design and Implementation Constraints

2.2.1 Hardware constraints

- **For running ASP.NET core on Windows 10 Pro**:
  - Processor: x86 or x64
  - RAM: 512Mb
  - Disk Space: 4.5Gb
- **For development Environment**
  - Microsoft Visual Studio 2015 Update 3 (minimum), 2017 version latest (15.6.4), 2019.
  - .NET Core: Minimum .NET core 3.0 upto .NET 3.0.
  - Command Line (Optional. Necessary for command-line deployment)
  - SQL Server 2008 Express (optional)

2.2.2 Sofware constraints

- **Windows OS**: The following Windows versions are supported with .NET Core 3.1

(A + symbol represents the minimum version)

OS	Version	Architectures
Windows Client	8.1	x64, x86
Windows 10 Client	Version 1609+	x64, x86
Windows Server	2012 R2+	x64, x86
Nano Server	Version 1803+	x64, ARM32

Table 2. Windows version support .NET Core 3.1

- **Visual Studio**: the following table describes the minimum required version of Visual Studio based on the target .NET Core SDK version.

.NET Core SDK version	Visual Studio version
3.1	Visual Studio 2019 version 16.4 or higher.

Table 3. Visual Studio version

- **Google Chrome browser**: Version 84.0.4147.125 (Official Build) (64-bit)
- **Balsamiq Mockups**: Version 3.5.17
- **Bootstrap Studio**: Version 5.2.1

2.2.3 Database constraints

- **NOT NULL**: Ensures that a column cannot have a NULL value
- **UNIQUE**: Ensures that all values in a column are different
- **PRIMARY KEY**: A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- **FOREIGN KEY**: Uniquely identifies a row/record in another table
- **CHECK**: Ensures that all values in a column satisfies a specific condition
- **DEFAULT**: Sets a default value for a column when no value is specified
- **INDEX**: Used to create and retrieve data from the database very quickly

2.2.4 Design constraints

- **Lack of nesting**: The constraint here is therefore that HTML be able to be mapped into a sequence of paragraphs of styled text, and that if that text is edited that the editor should be able to map the sequence of styles back onto a sequence of elements in a well-defined way. This allows some limited trivial nesting (eg LI within UL) but no general nesting, as a finite and small set of styles is used. In particular, the styles are not parameterized by the nesting level.
- **Logical Markup**: It is required that HTML be a common language between all platforms. This implies no device-specific markup, or anything which requires control over fonts or colors, for example. This is in keeping with the SGML ideal. (w3, n.d.)

2.2.5 Programming constraints

Constraints inform the compiler about the capabilities a type of argument must-have. Without any constraints, the type of argument could be any type. The compiler can only assume the members of the System. Object, which is the ultimate base class for any.
The following table lists the types of constraints:

Constraint	Description
where T : struct	The type argument must be a non-nullable value type. For information about nullable value types, see Nullable value types. Because all value types have an accessible parameterless constructor, the struct constraint implies the new() constraint and can't be combined with the new() constraint. You can't combine the struct constraint with the unmanaged constraint.
where T : class	The type argument must be a reference type. This constraint applies also to any class, interface, delegate, or array type. In a nullable context in C# 8.0 or later, T must be a non-nullable reference type.
where T : class?	The type argument must be a reference type, either nullable or non-nullable. This constraint applies also to any class, interface, delegate, or array type.
where T : notnull	The type argument must be a non-nullable type. The argument can be a non-nullable reference type in C# 8.0 or later, or a non-nullable value type.
where T : unmanaged	The type argument must be a non-nullable unmanaged type. The unmanaged constraint implies the struct constraint and can't be combined with either the struct or new() constraints.
where T : new()	The type argument must have a public parameterless constructor. When used together with other constraints, the new() constraint must be specified last. The new() constraint can't be combined with the struct and unmanaged constraints.
where T : <base class name>	The type argument must be or derive from the specified base class. In a nullable context in C# 8.0 and later, T must be a non-nullable reference type derived from the specified base class.
where T : <base class name>?	The type argument must be or derive from the specified base class. In a nullable context in C# 8.0 and later, T may be either a nullable or non-nullable type derived from the specified base class.
where T : U	The type argument supplied for T must be or derive from the argument supplied for U. In a nullable context, if U is a non-nullable reference type, T must be non-nullable reference type. If U is a nullable reference type, T may be either nullable or non-nullable.

Table 4. Programming constraints. (microsoft, n.d.)

2.2.6 Business constraints

- Usernames and Emails are unique instances.
- Category names are unique instances.
- Course names are unique instances.
- Administrators can only manage users such as training staff, trainers and trainees.
- The trainer can only view their courses and topics when assigned to the topics
- Training staff can only manage users such as trainers and trainees.

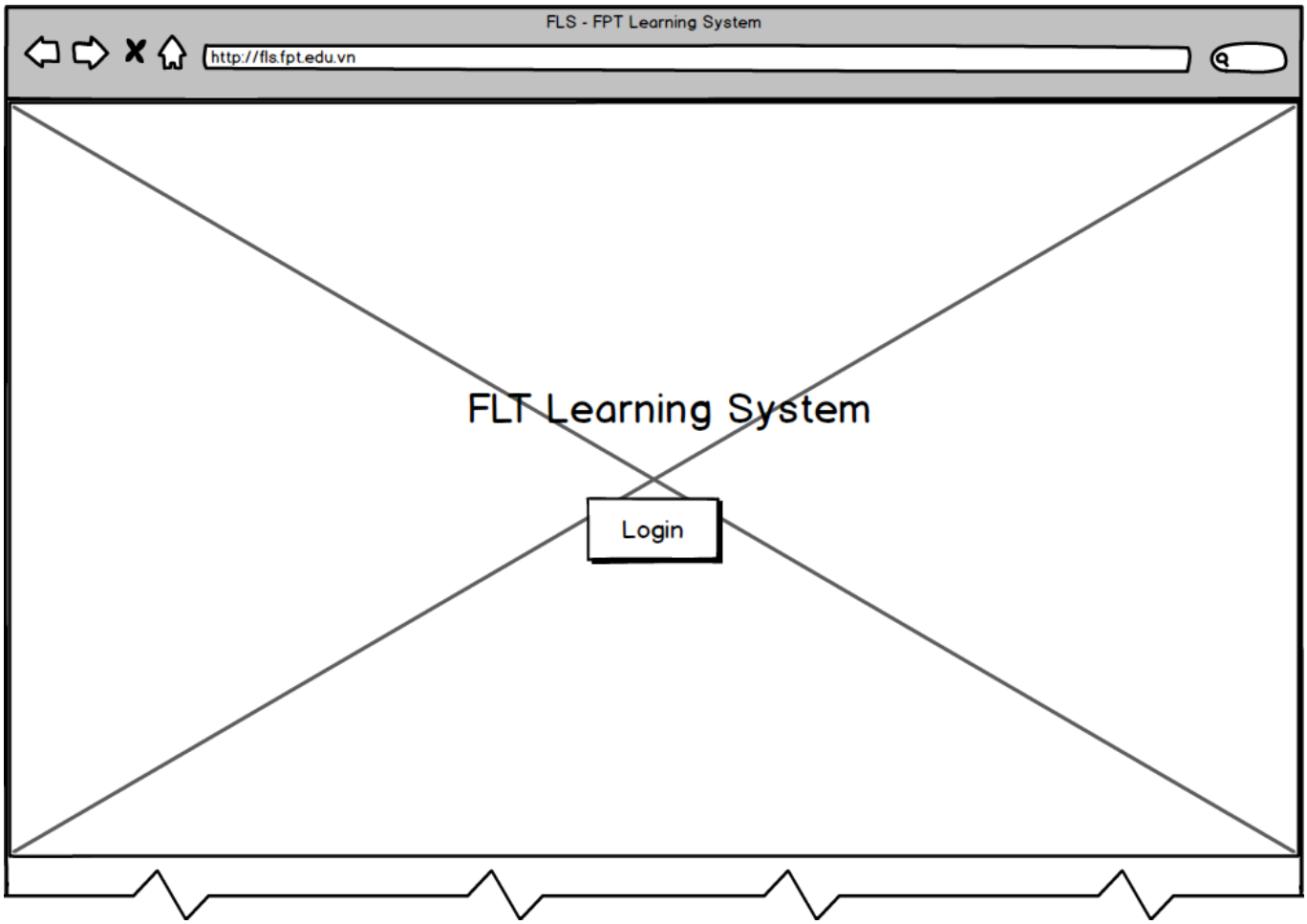
2.2.7 Security constraints

- **Web resource collection**: A list of URL patterns and HTTP that describe a set of resources to be protected. Web resource collections are discussed in Specifying a Web Resource Collection
- **Authorization constraint**: Specifies whether authentication is to be used and names the roles authorized to perform the constrained requests. For more information about authorization constraints.
- **User data constraint**: Specifies how data is protected when transported between a client and a server.

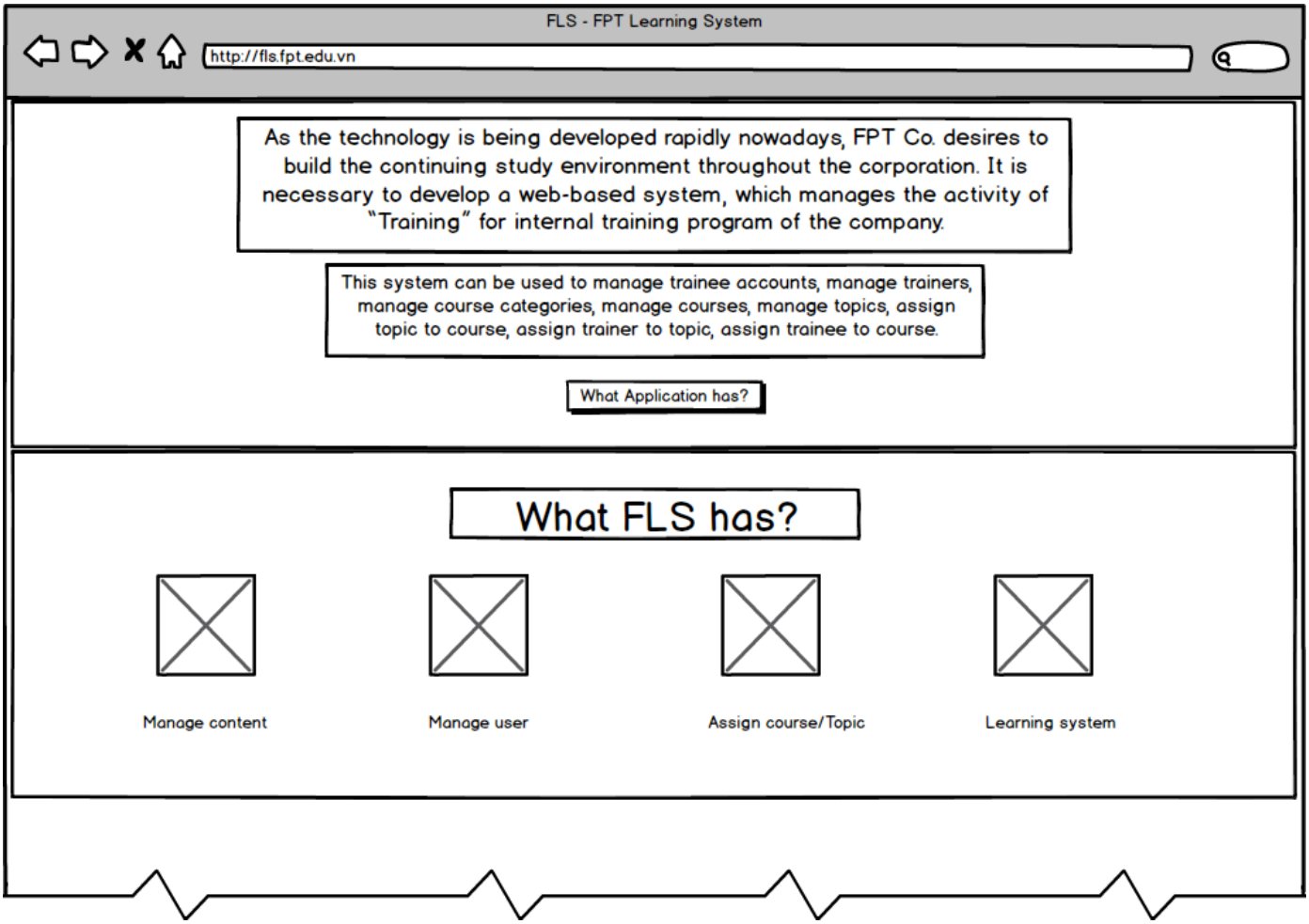
3 Specific Requirements

3.1 User interfaces

- Index – Unauthenticated

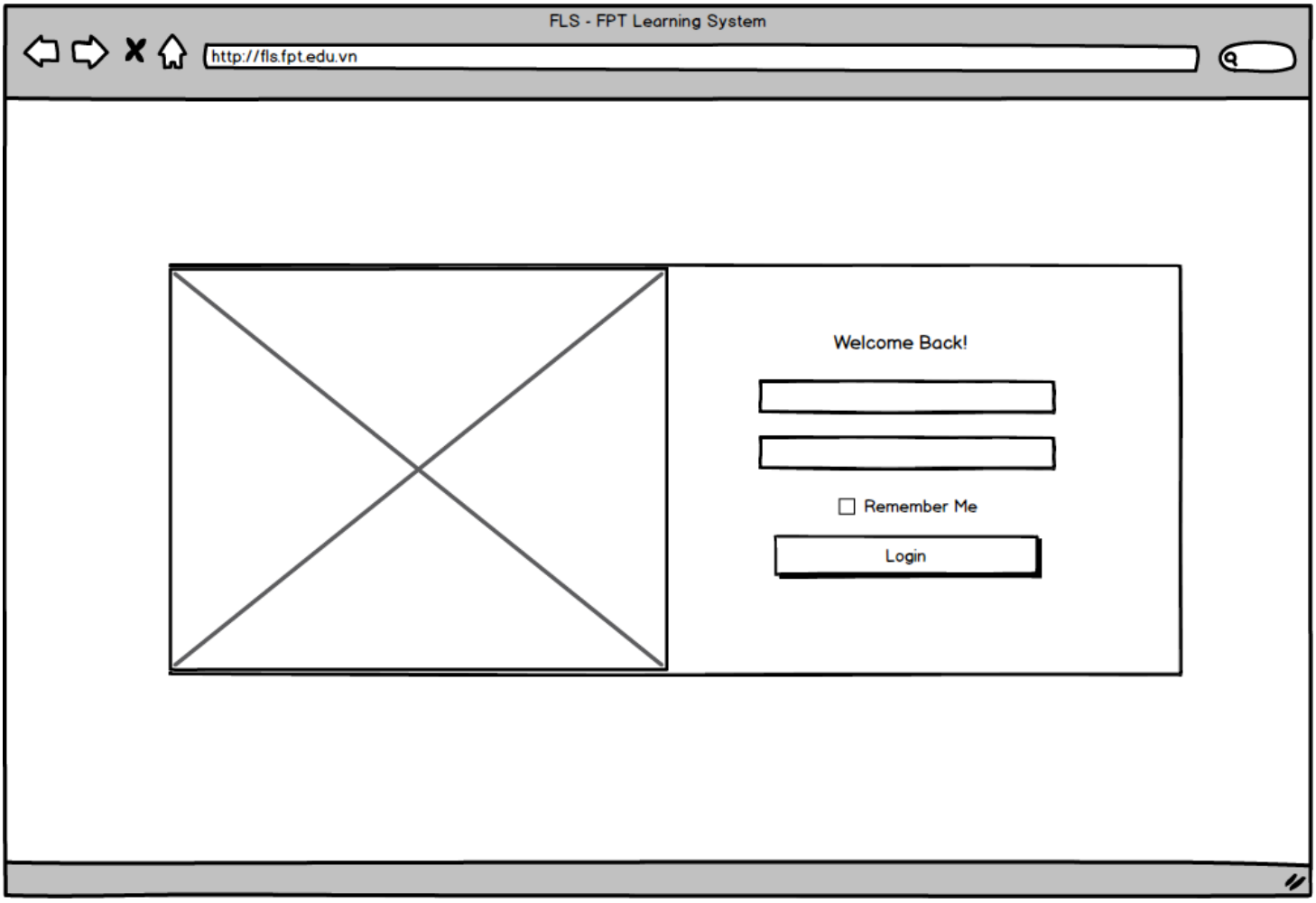


Picture 1. Index page for UnAuthenticated



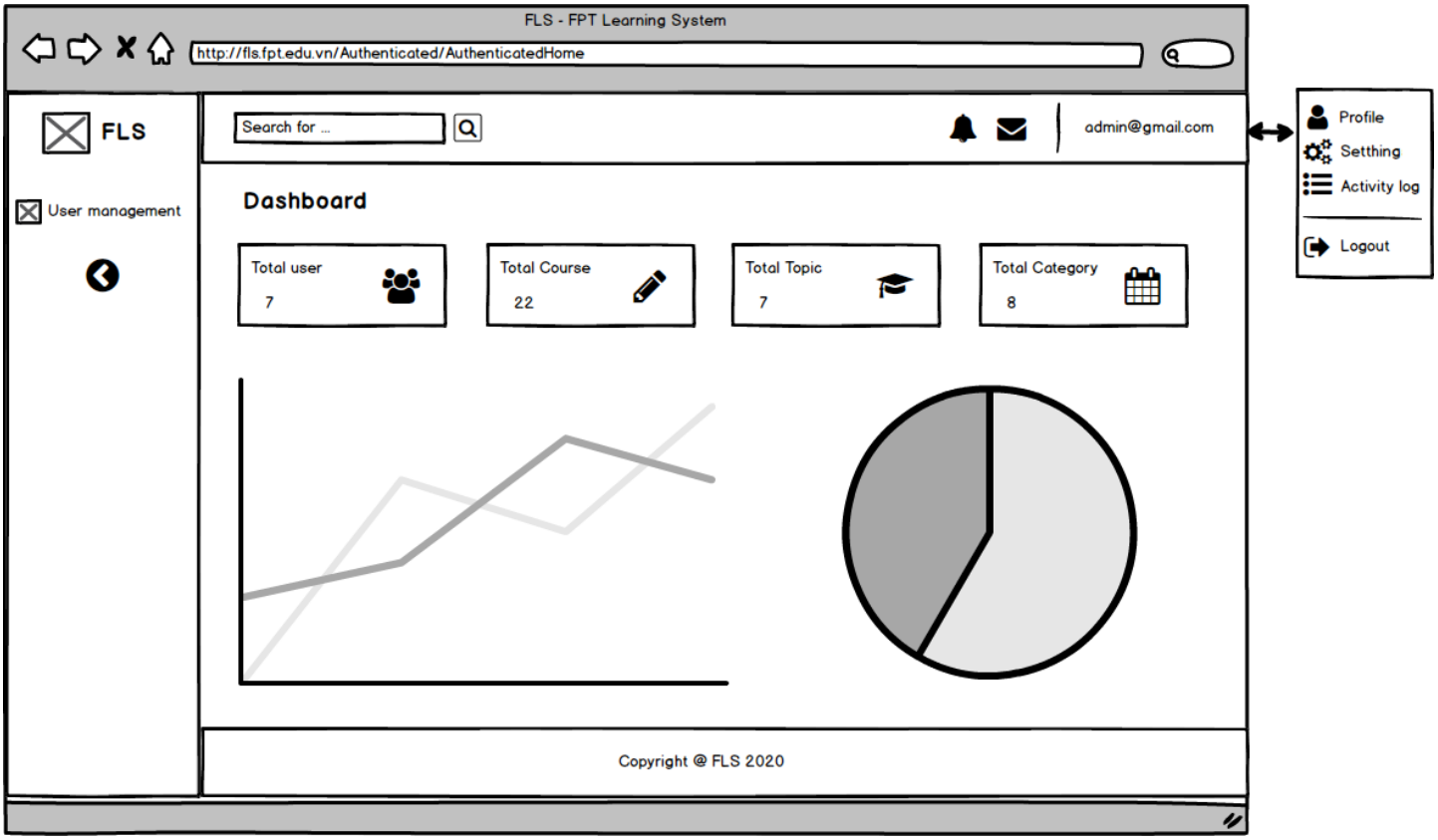
Picture 2. Index page for UnAuthenticated

- Login



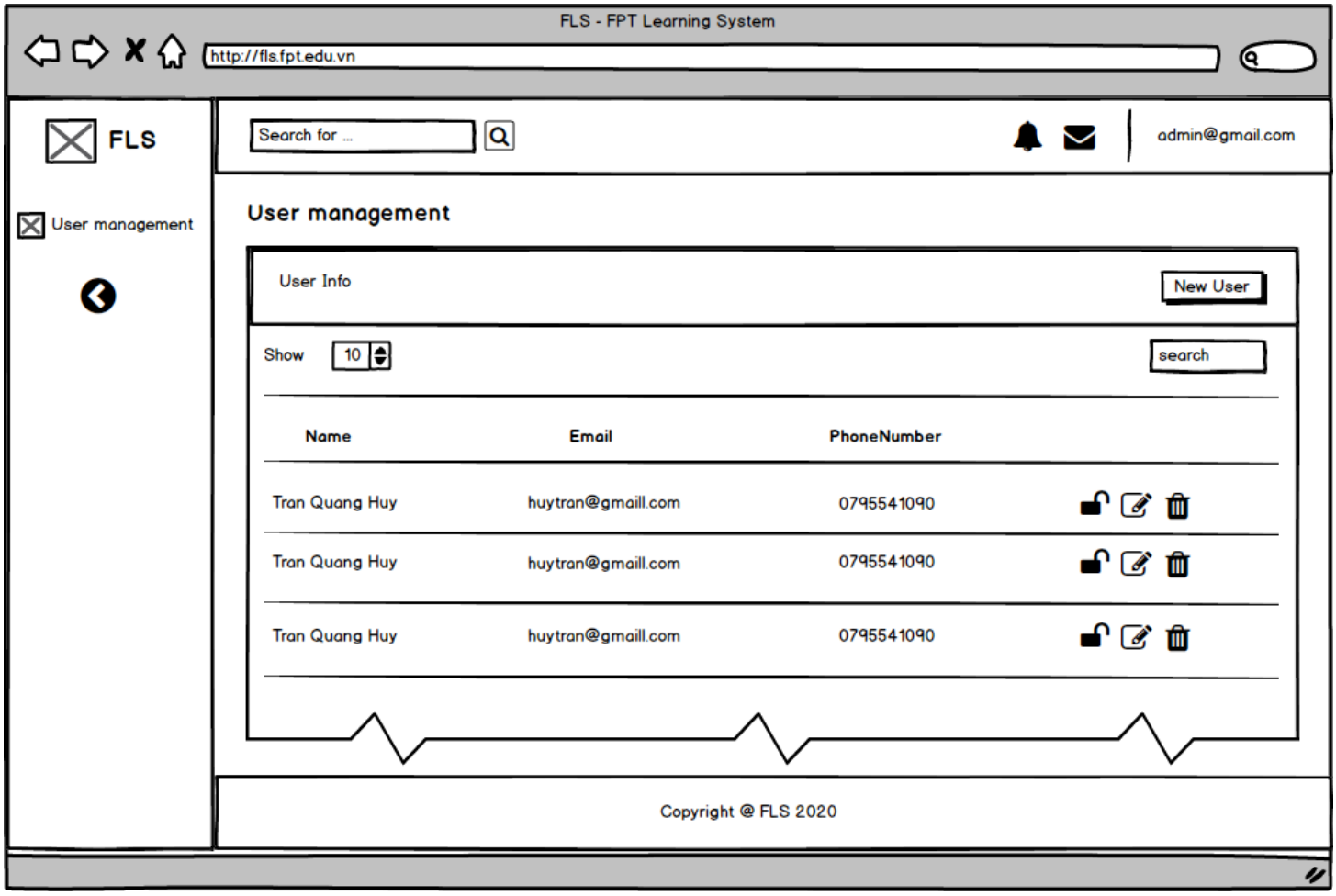
Picture 3. Login page

- Admin – Dashboard



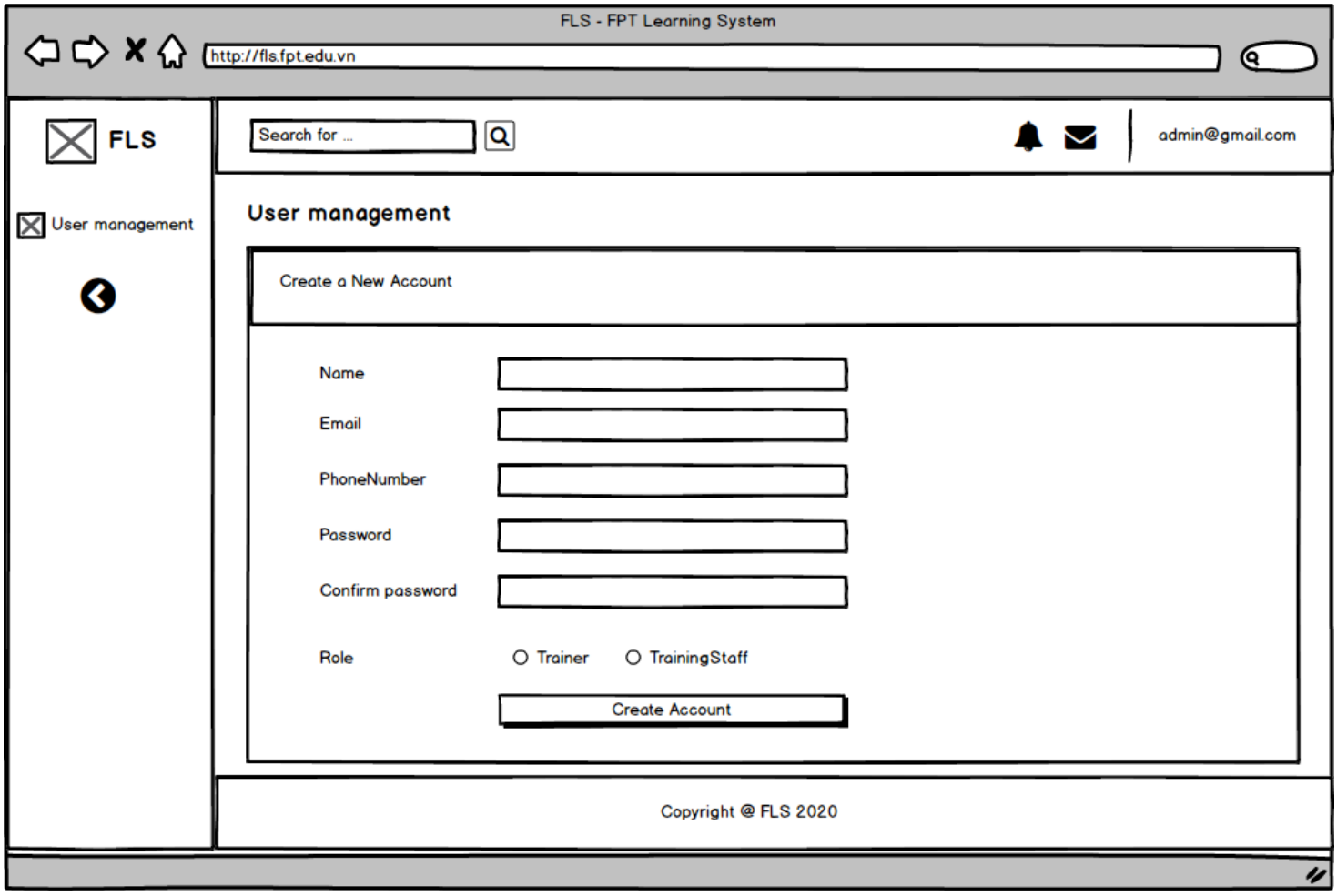
Picture 4. Admin - Dashboard

- Admin – User management – Index



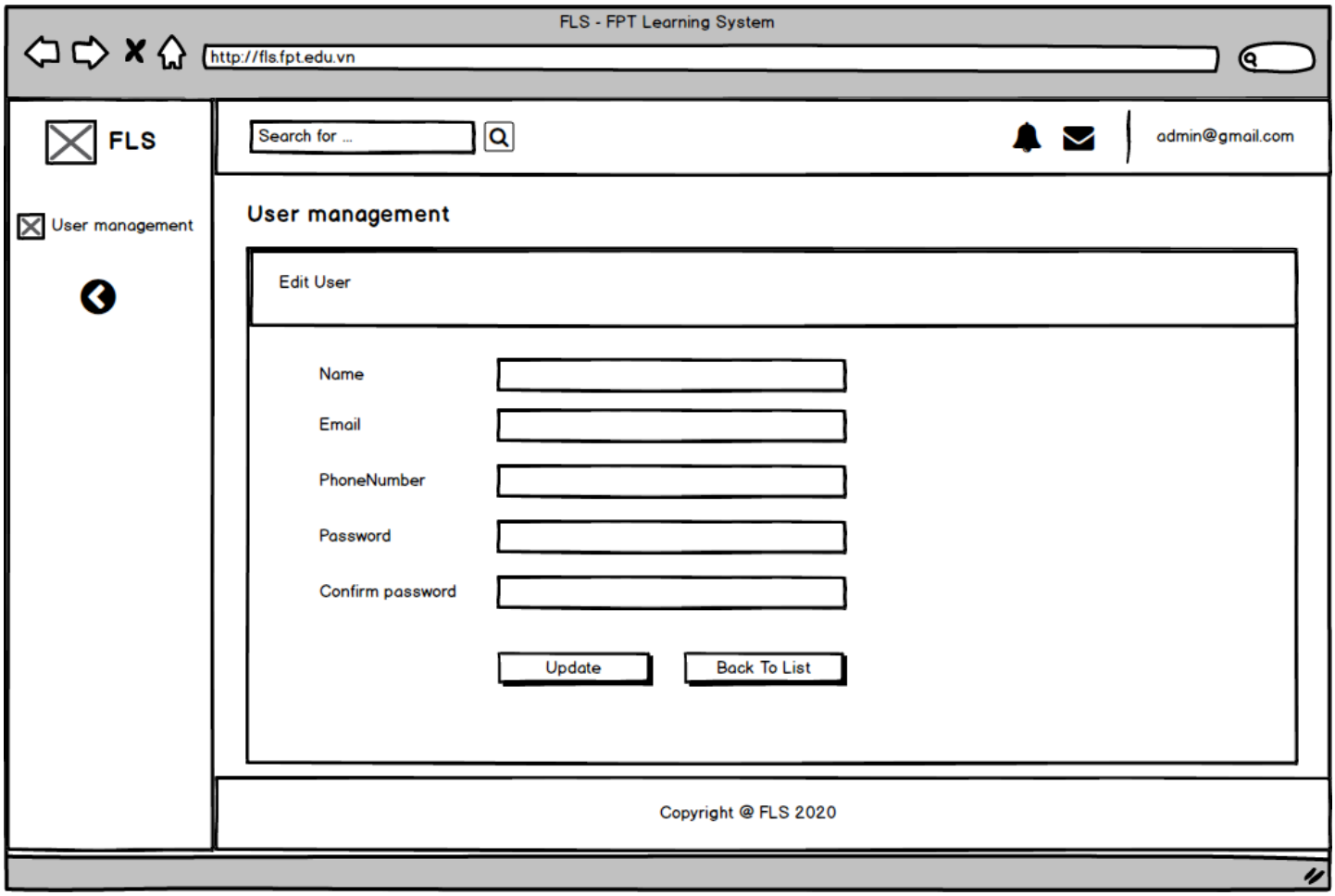
Picture 5. Admin - User management index page

- Admin – User management – Create



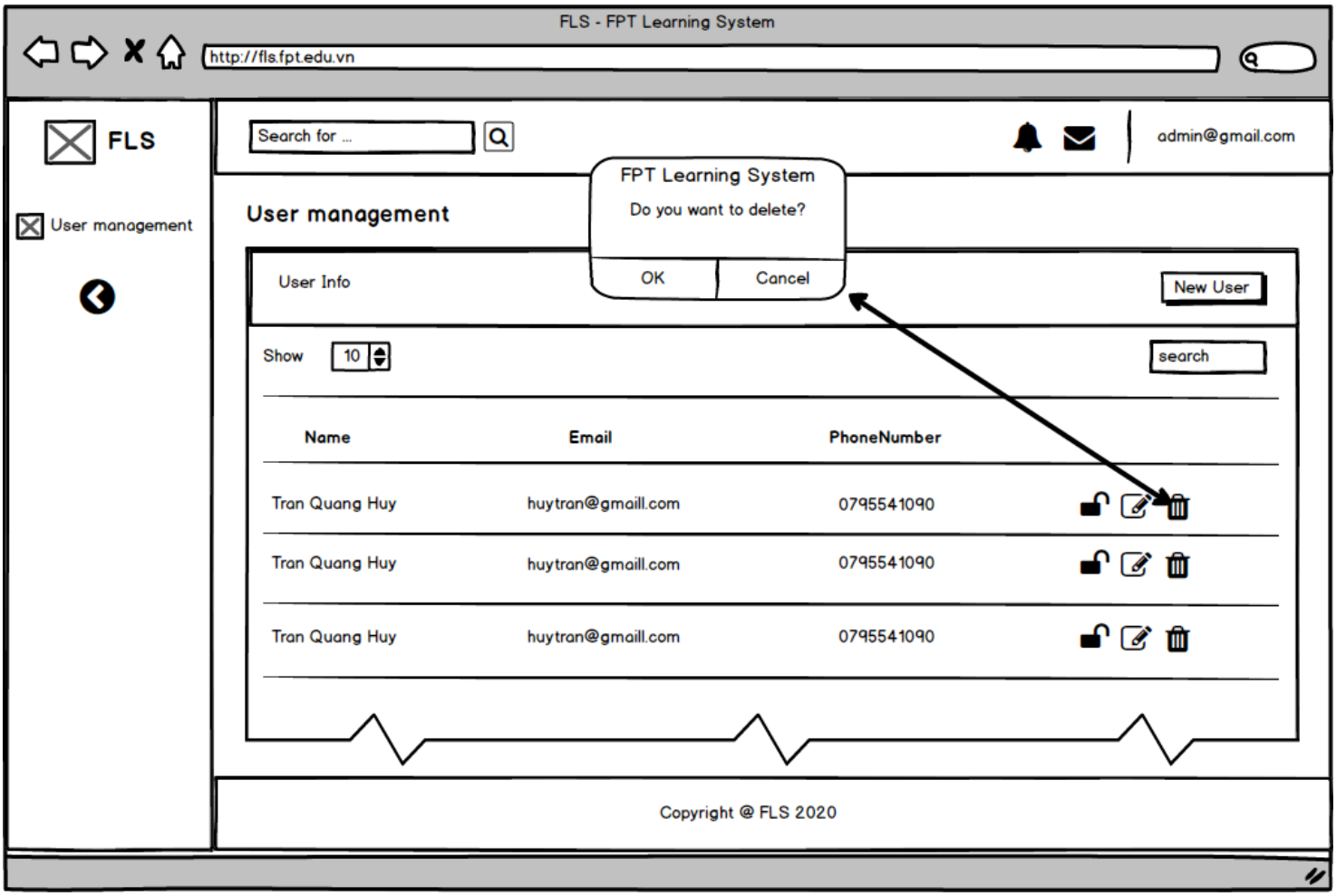
Picture 6. Admin - User management create page

- Admin – User management – Edit



Picture 7. Admin - User management edit page

- Admin – User management – Delete

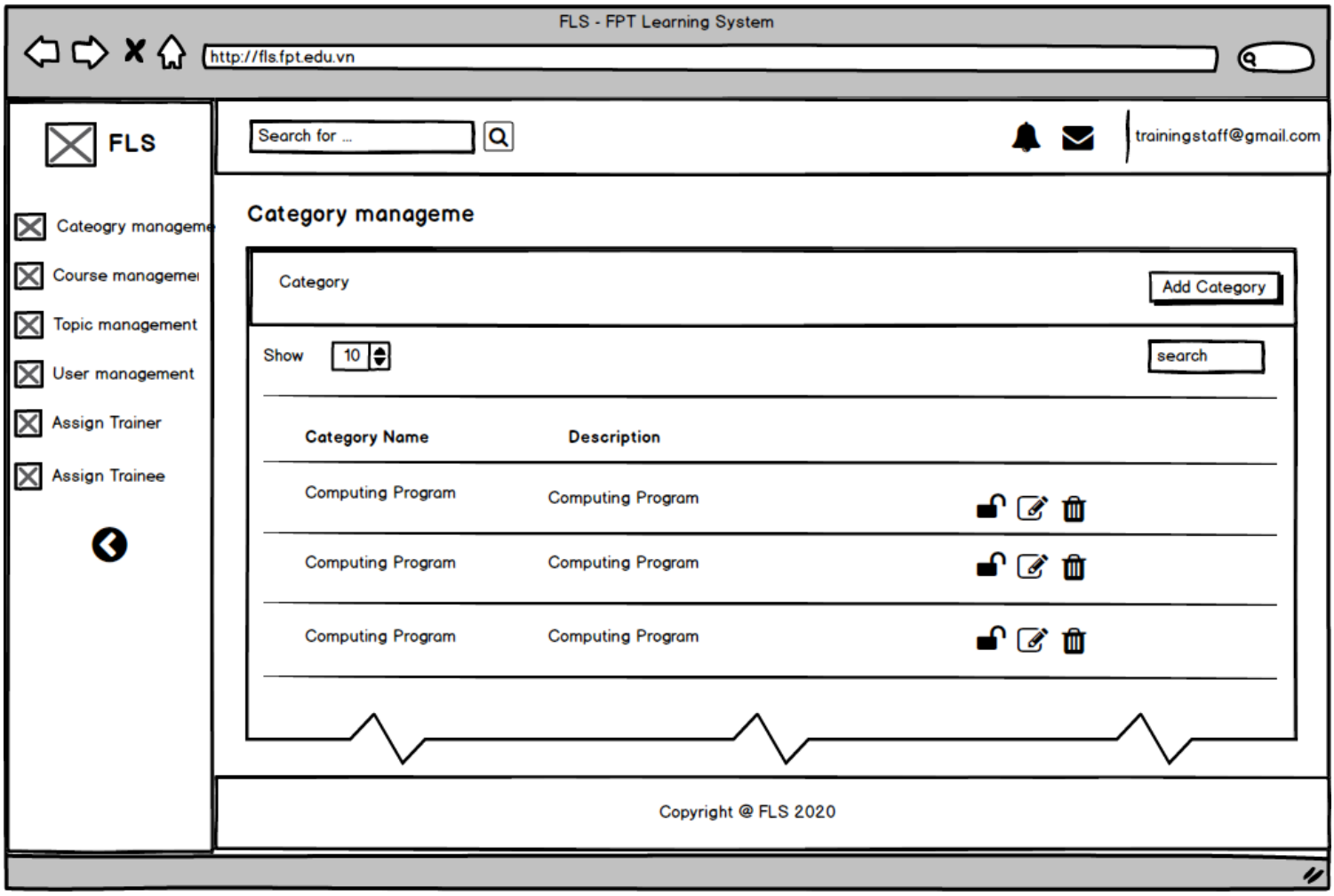


Picture 8. Admin - User management delete page

- TrainingStaff – Dashboard

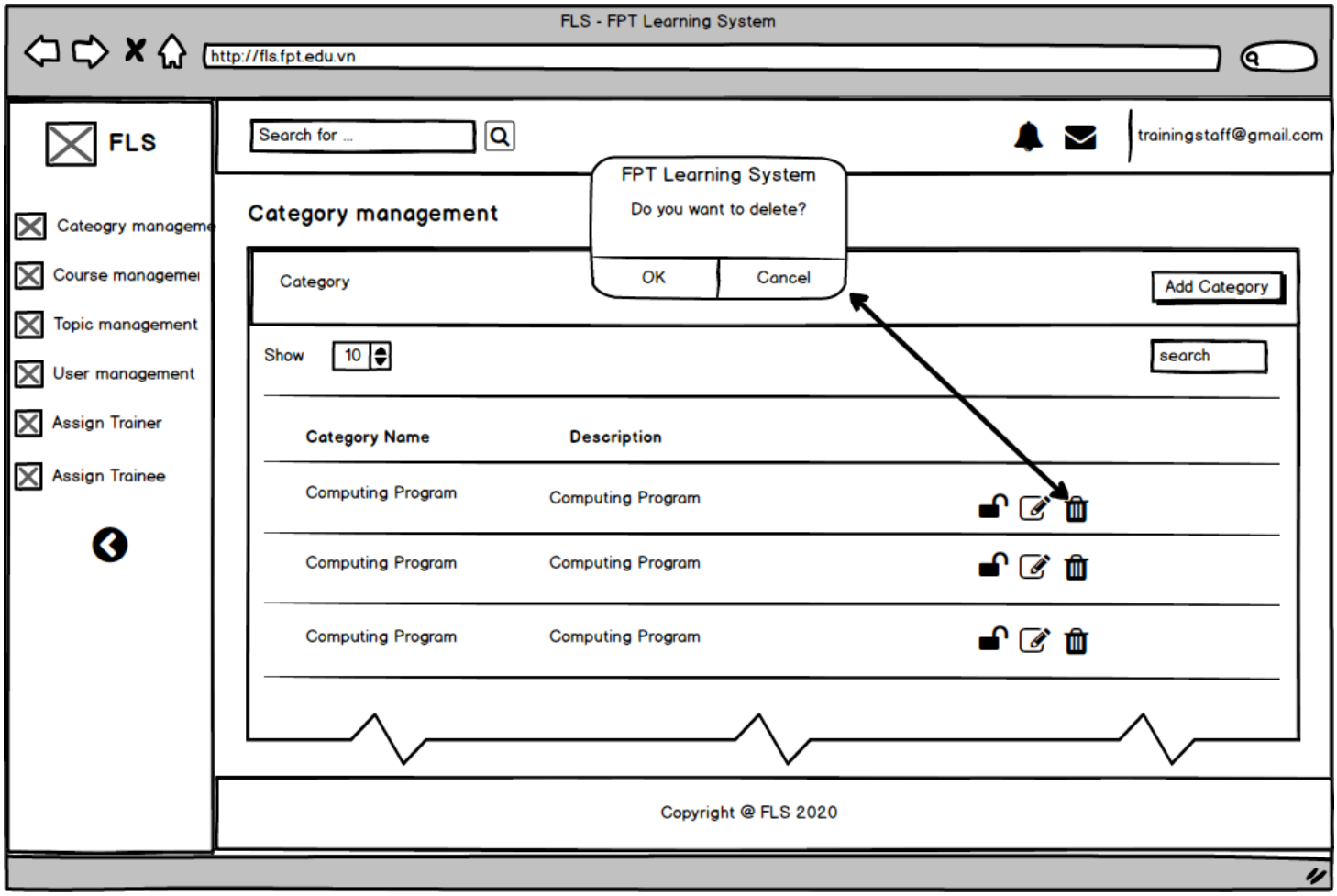
Picture 9. Training Staff dashboard

- TrainingStaff – Category management – Index



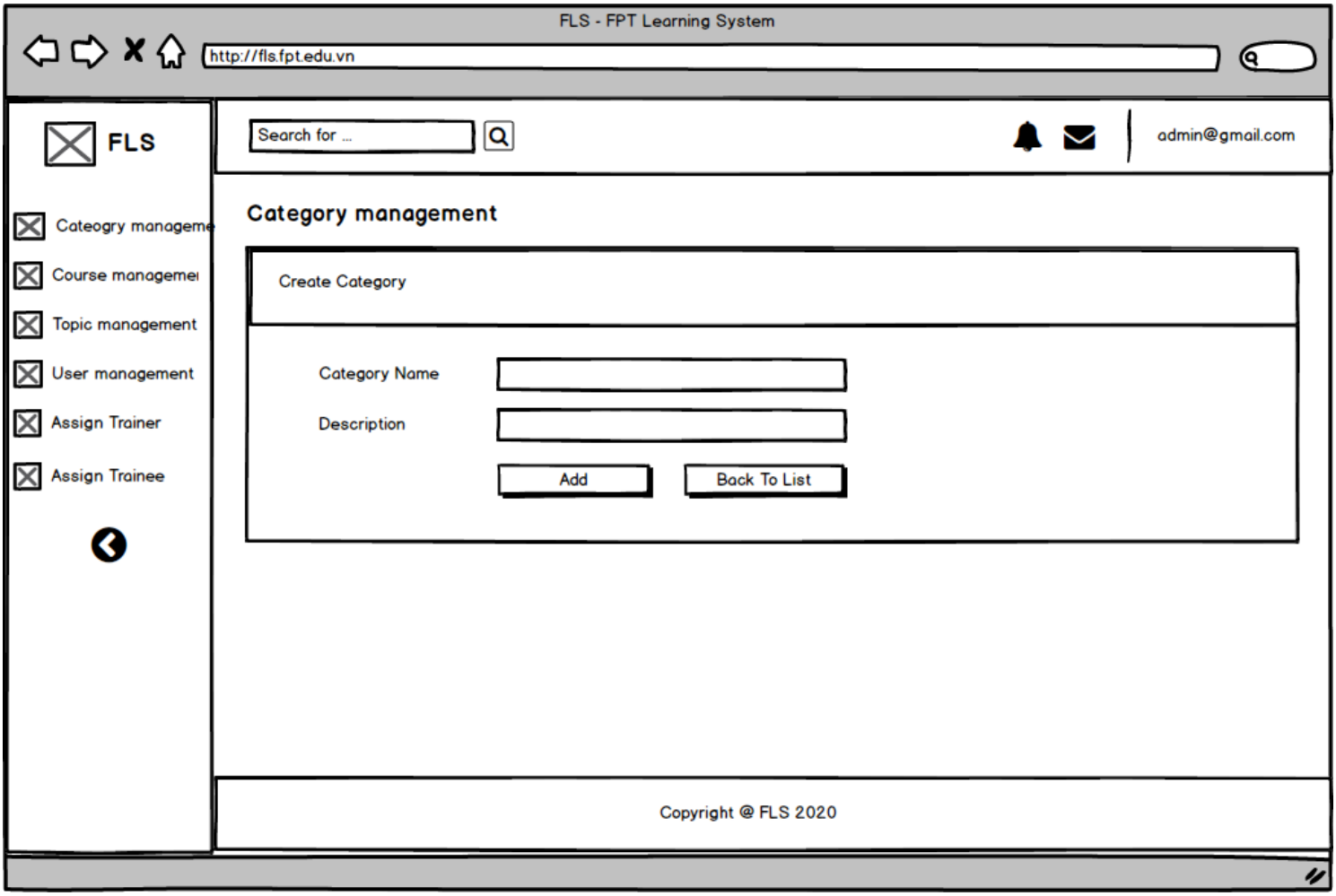
Picture 10. Training Staff - Category management index page

- TrainingStaff – Category management – Delete



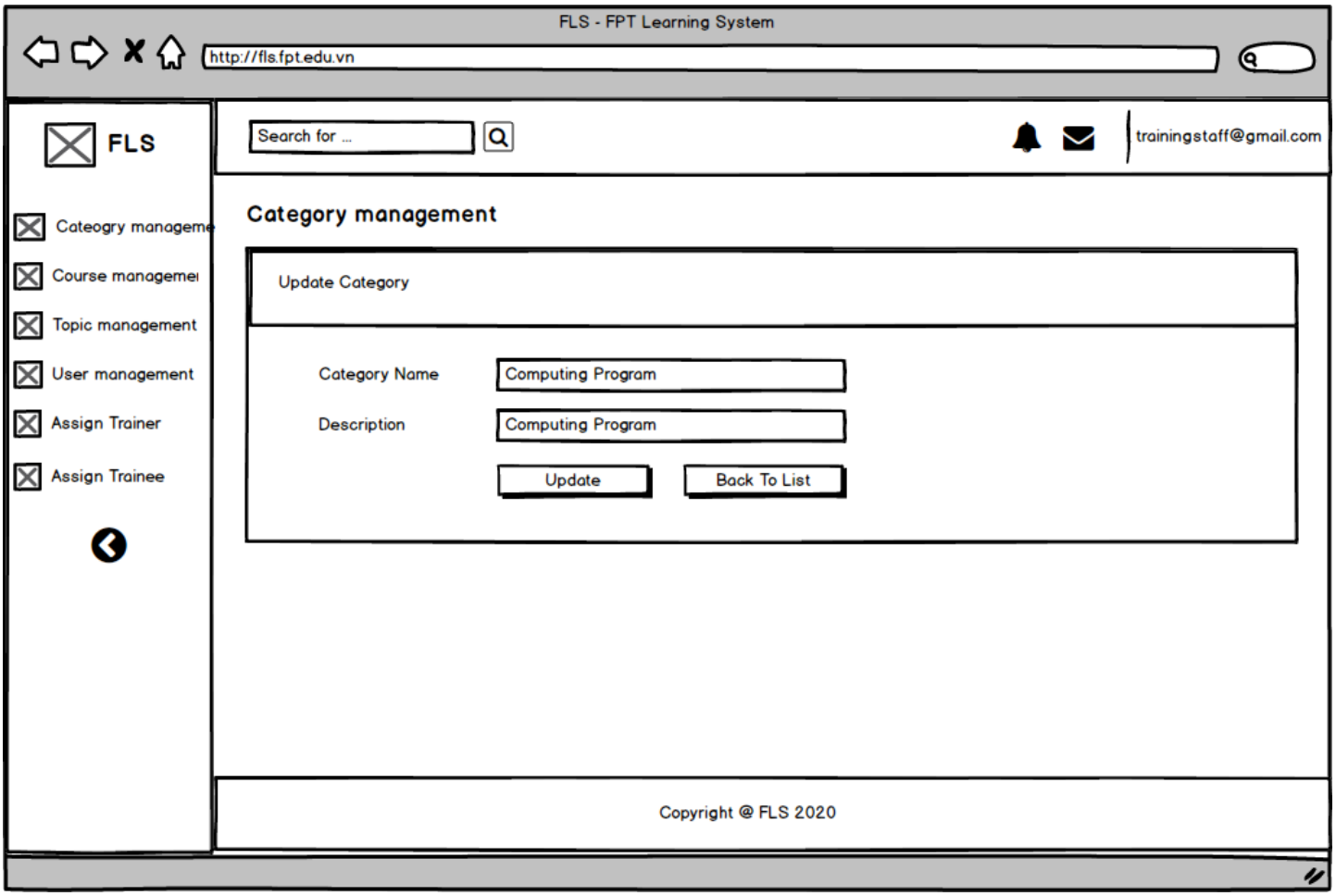
Picture 11. Training Staff - Category management delete page

- TrainingStaff – Category management – Create



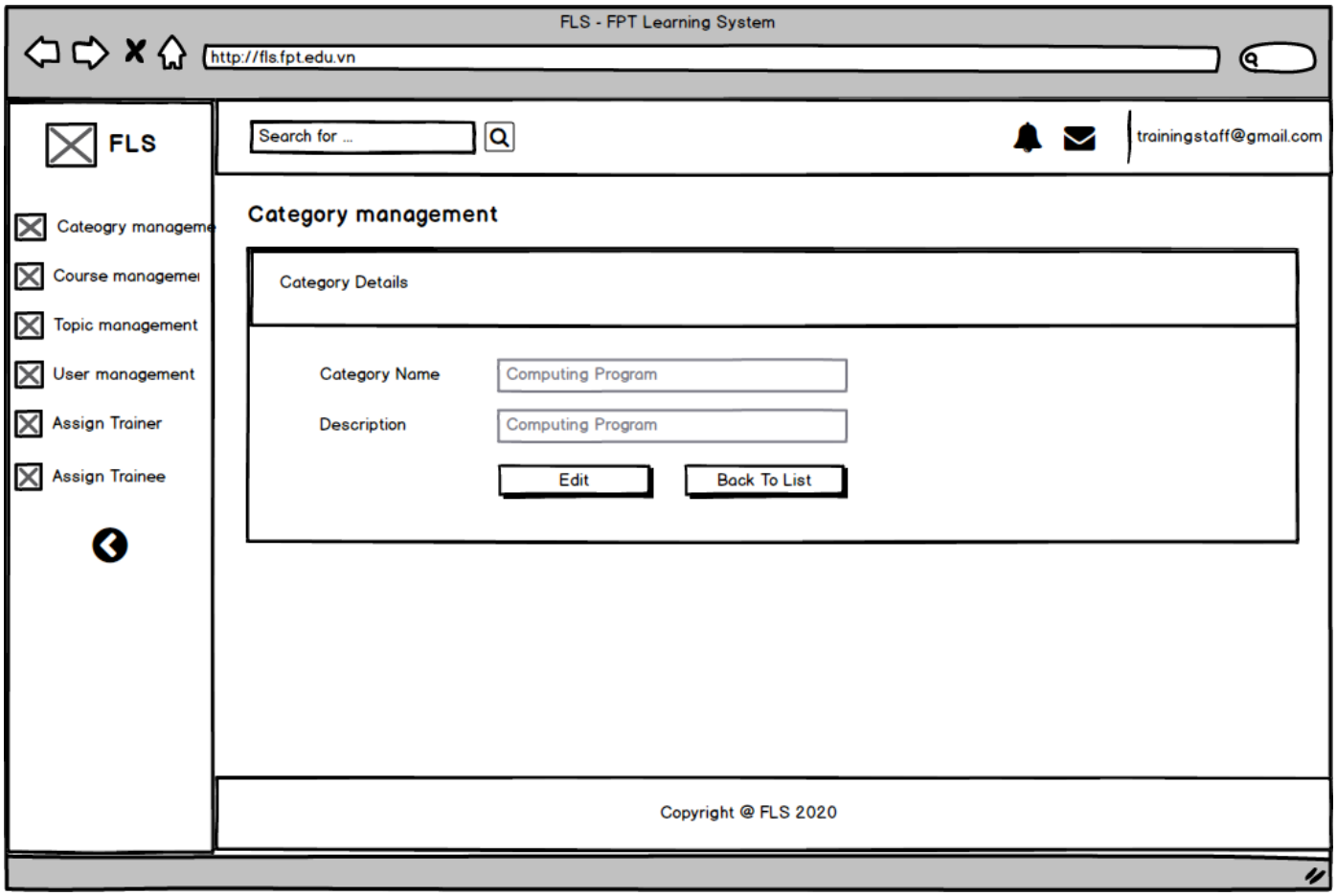
Picture 12. Training Staff - Category management create page

- TrainingStaff – Category management – Edit



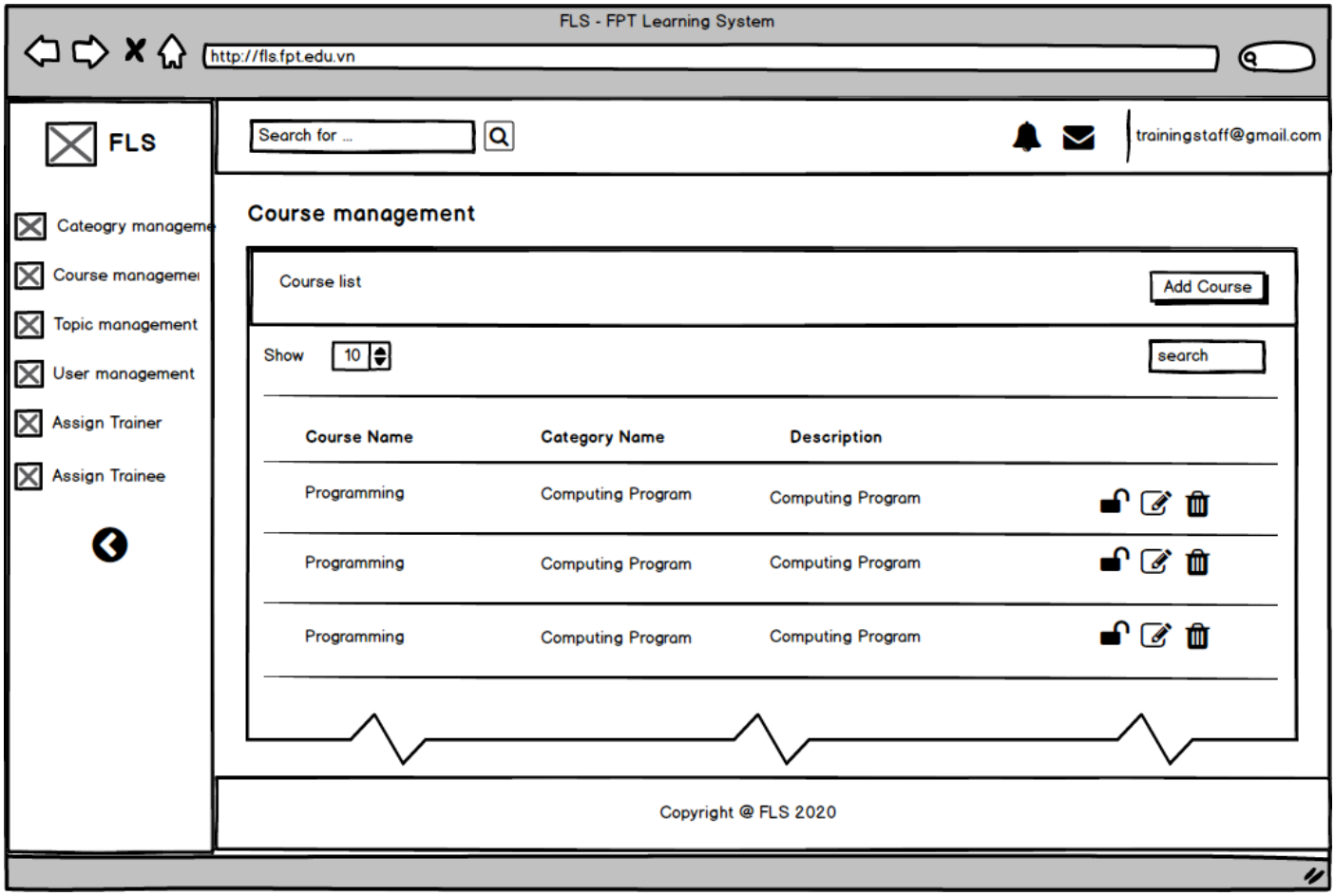
Picture 13. Training Staff - Category management edit page

- TrainingStaff – Category management – Details



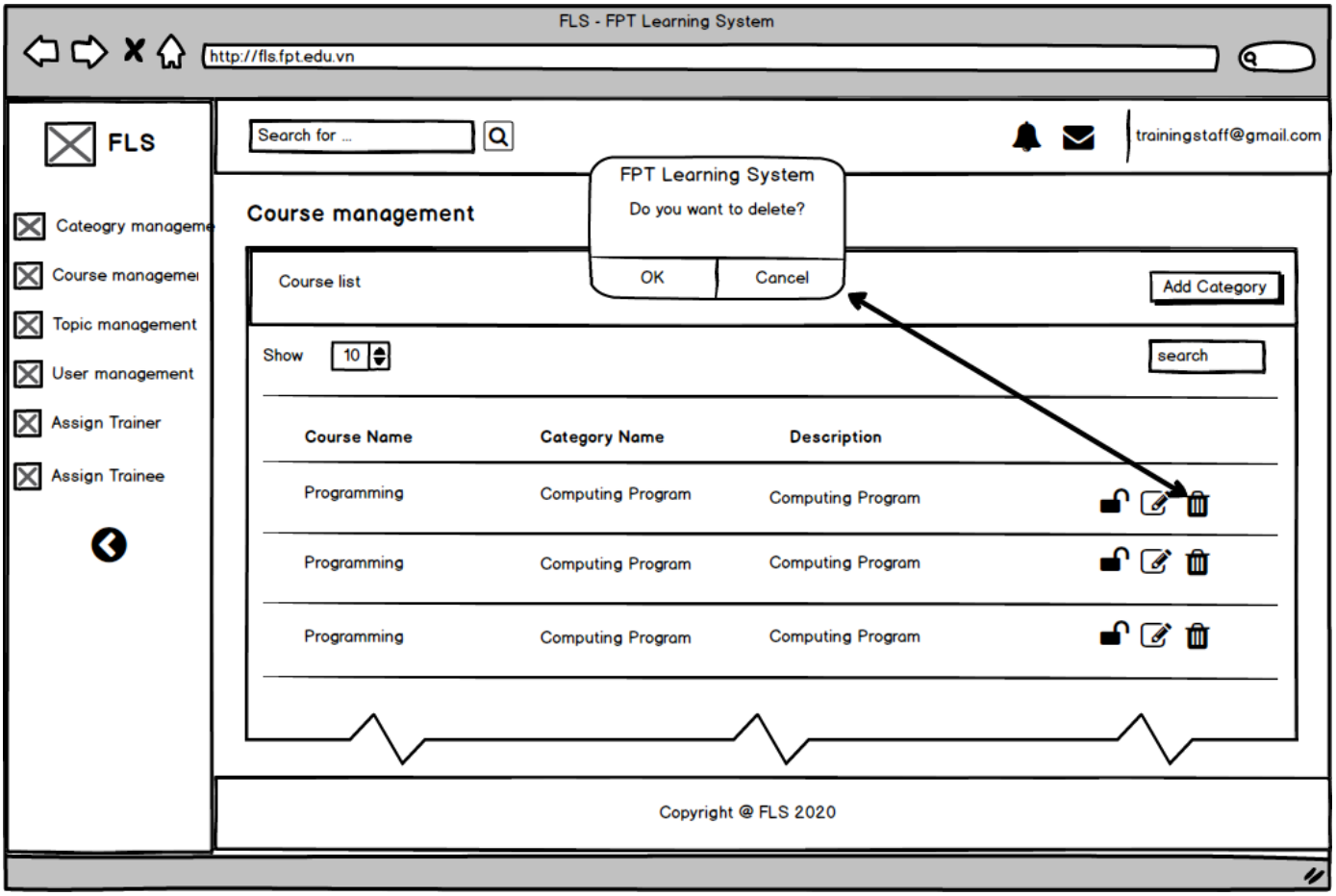
Picture 14. Training Staff - Category management details page

- TrainingStaff – Course management – Index



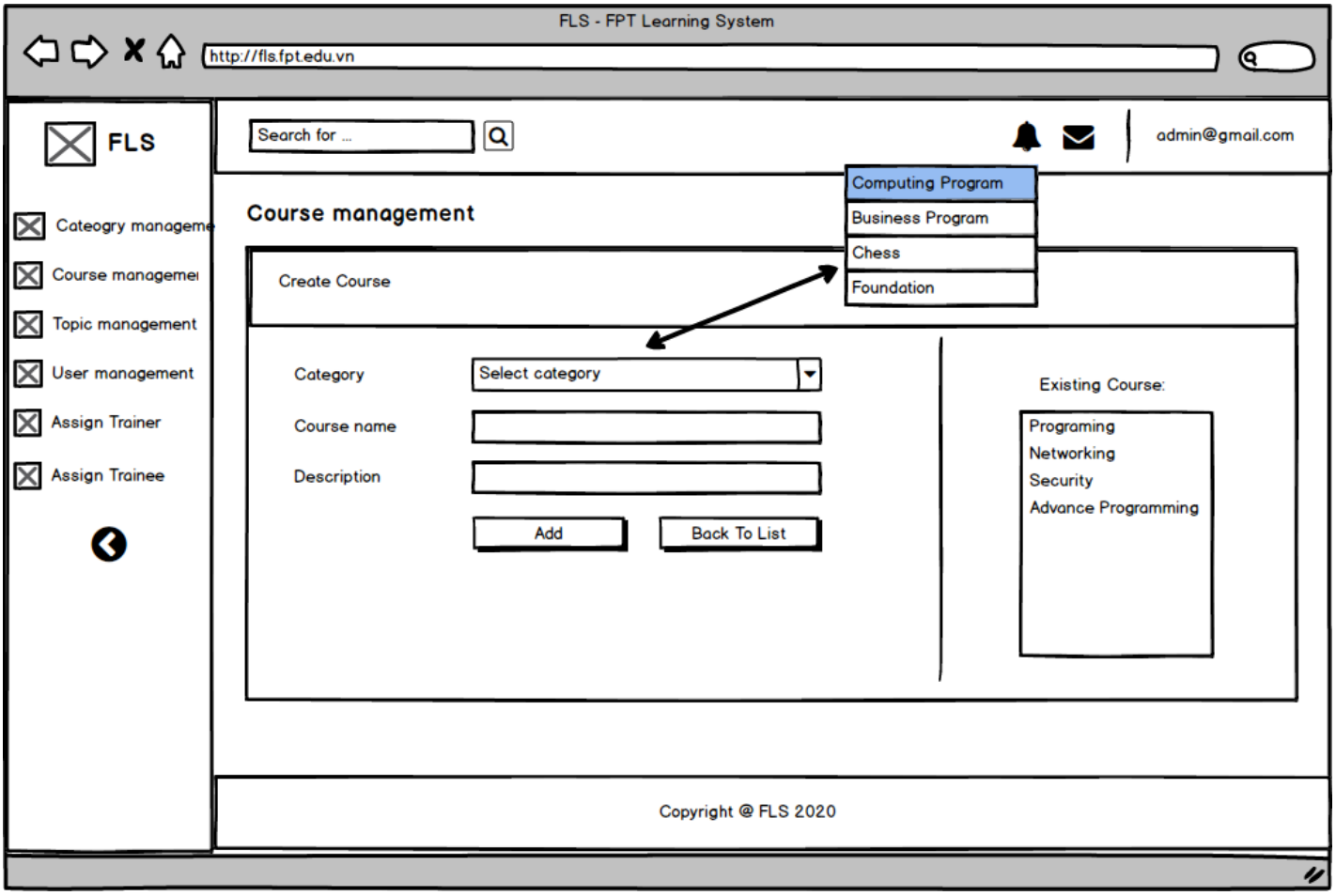
Picture 15. Training Staff - Course management index page

- TrainingStaff – Course management – Delete



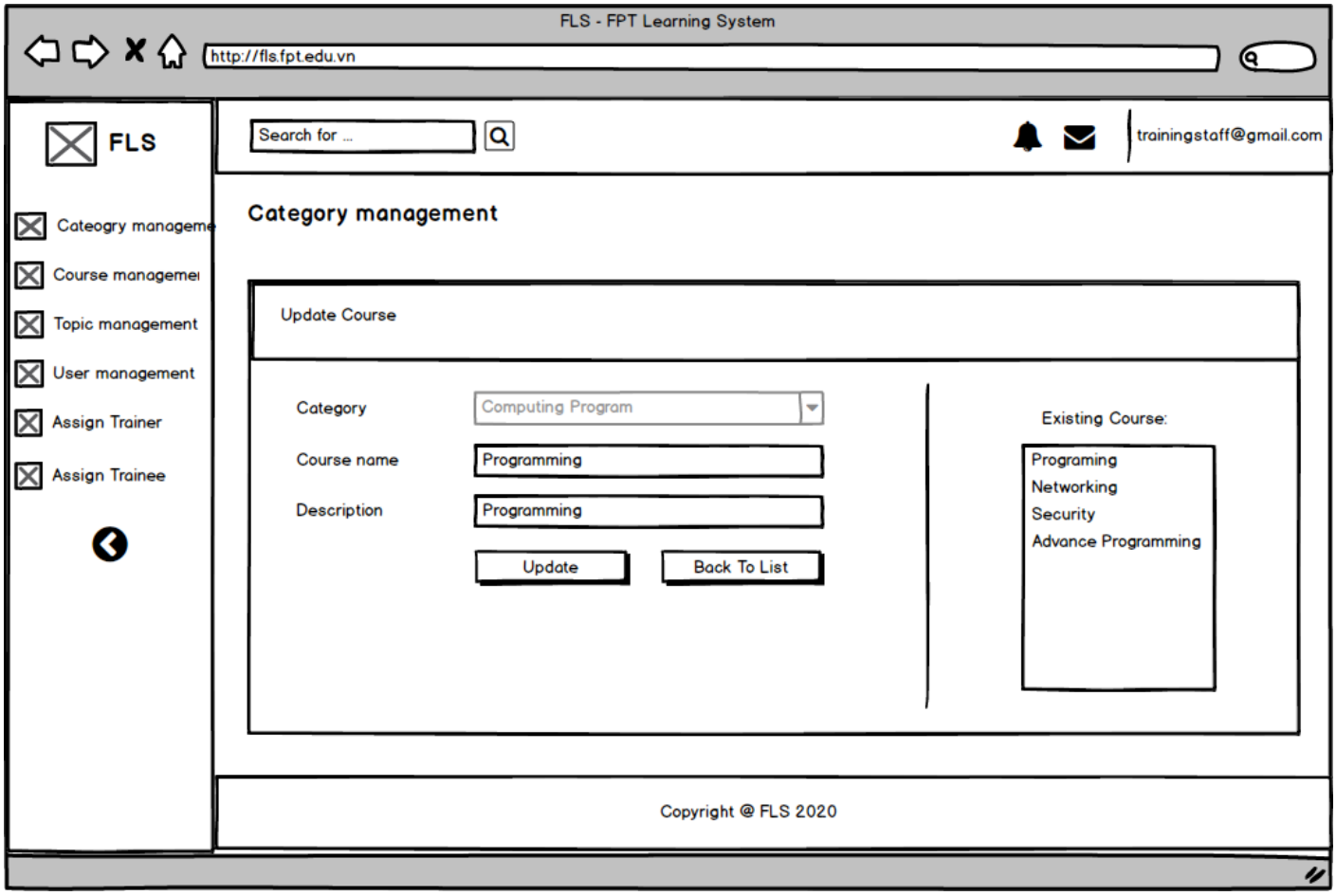
Picture 16. Training Staff - Course mângement delete page

- TrainingStaff – Course management – Create



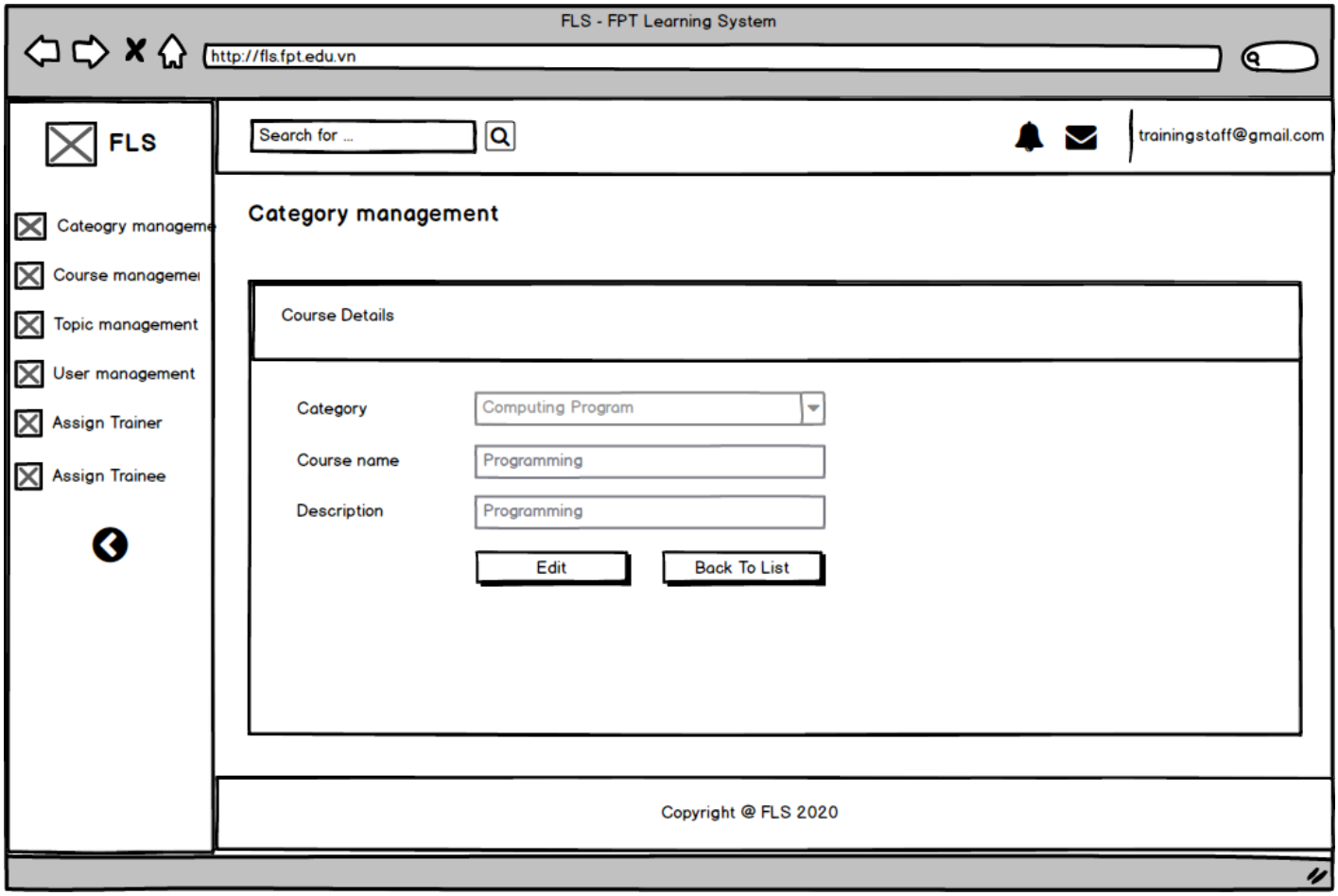
Picture 17. Training Staff - Course management create page

- TrainingStaff – Course management – Edit



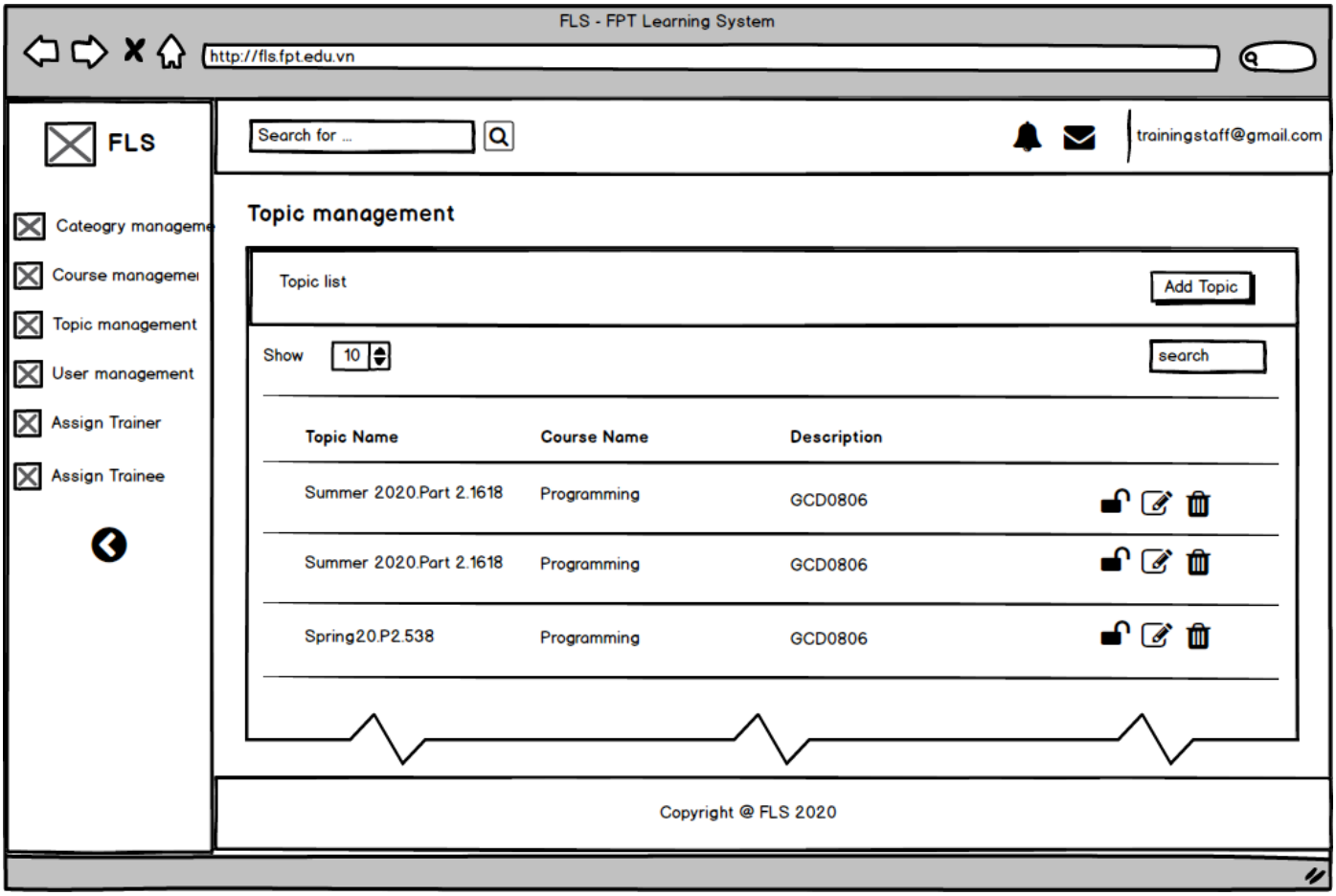
Picture 18. Training Staff - Course management edit page

- TrainingStaff – Course management – Details



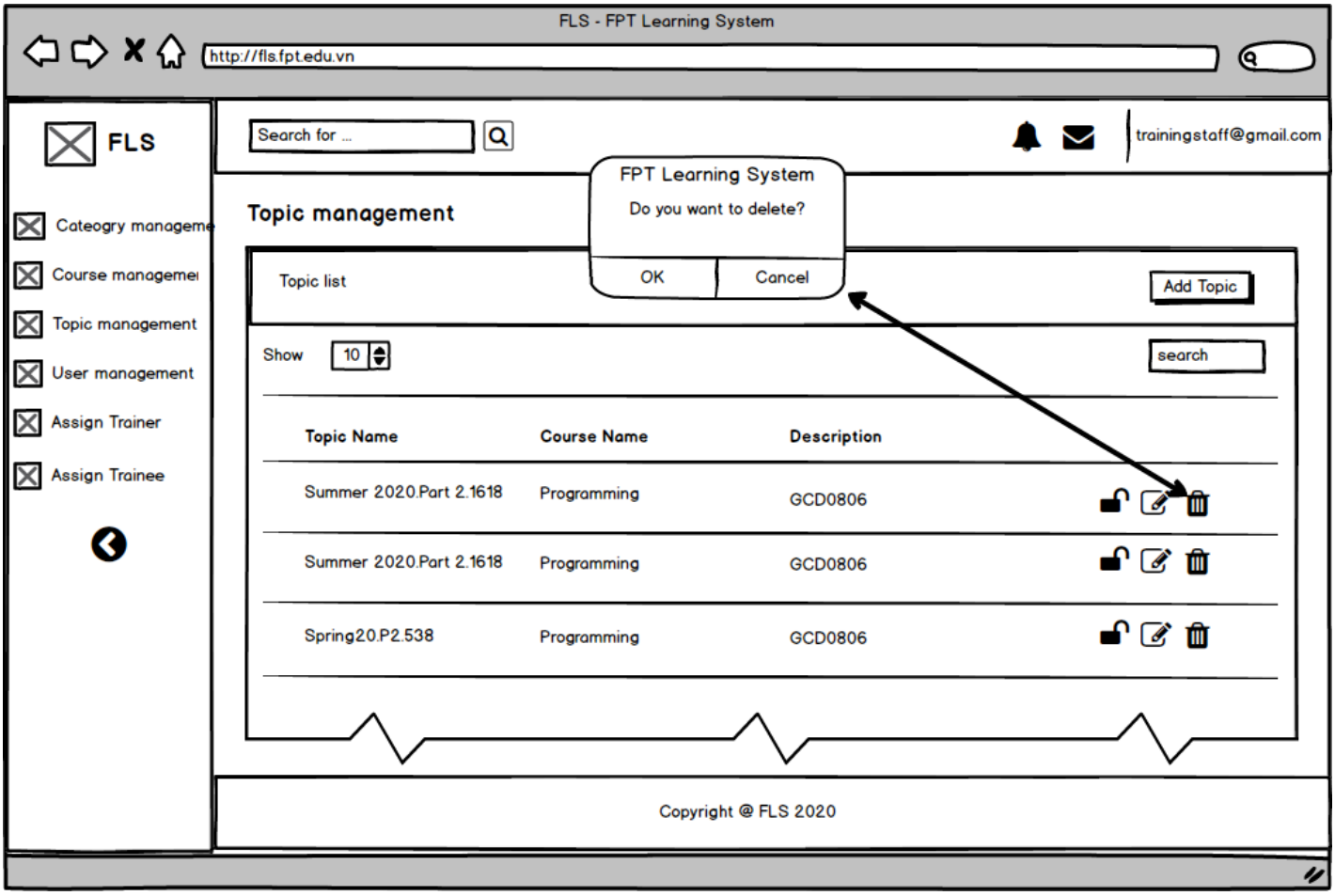
Picture 19. Training Staff - Course management details page

- TrainingStaff – Topic management – Index



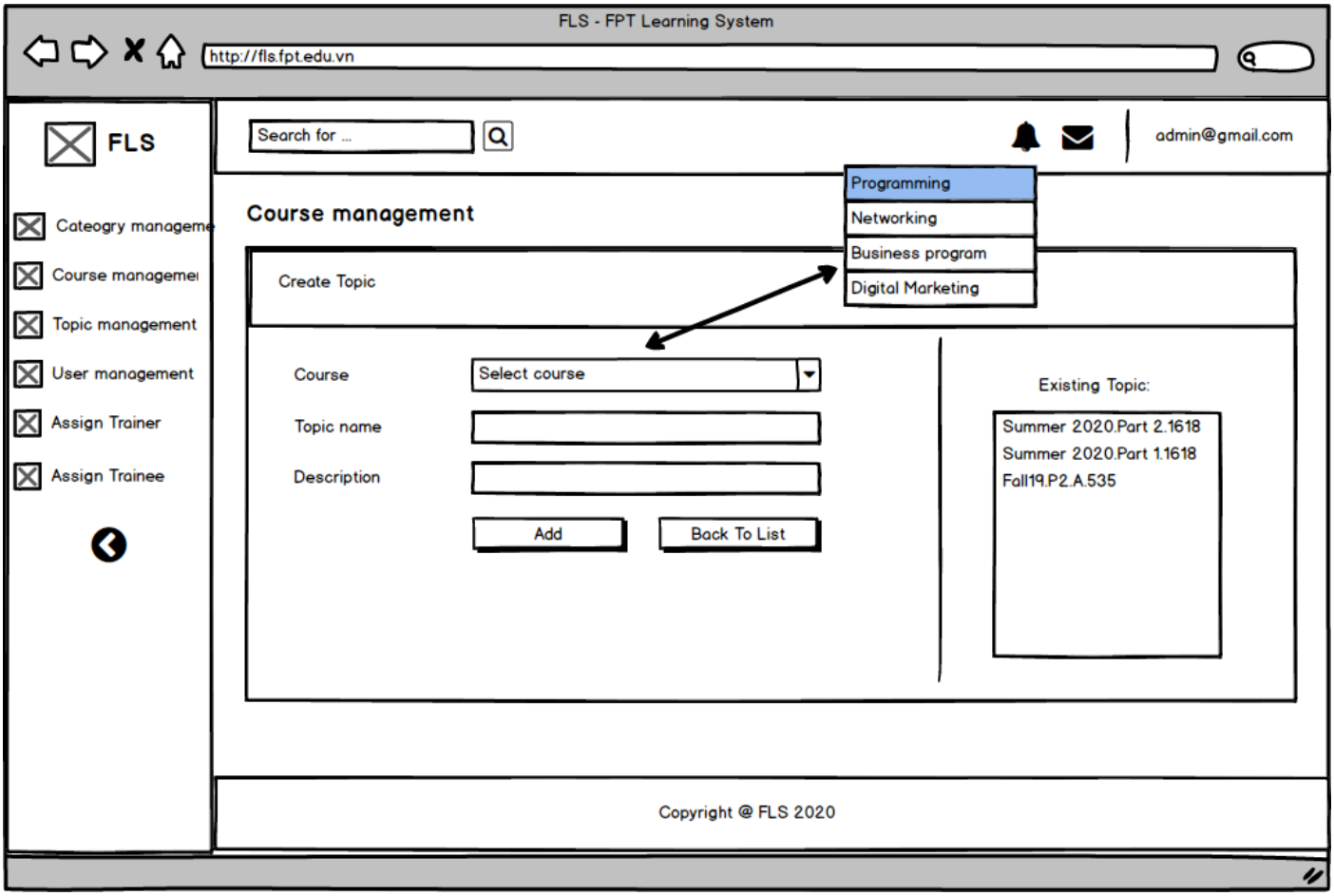
Picture 20. Training Staff - Topic management index page

- TrainingStaff – Topic management – Delete



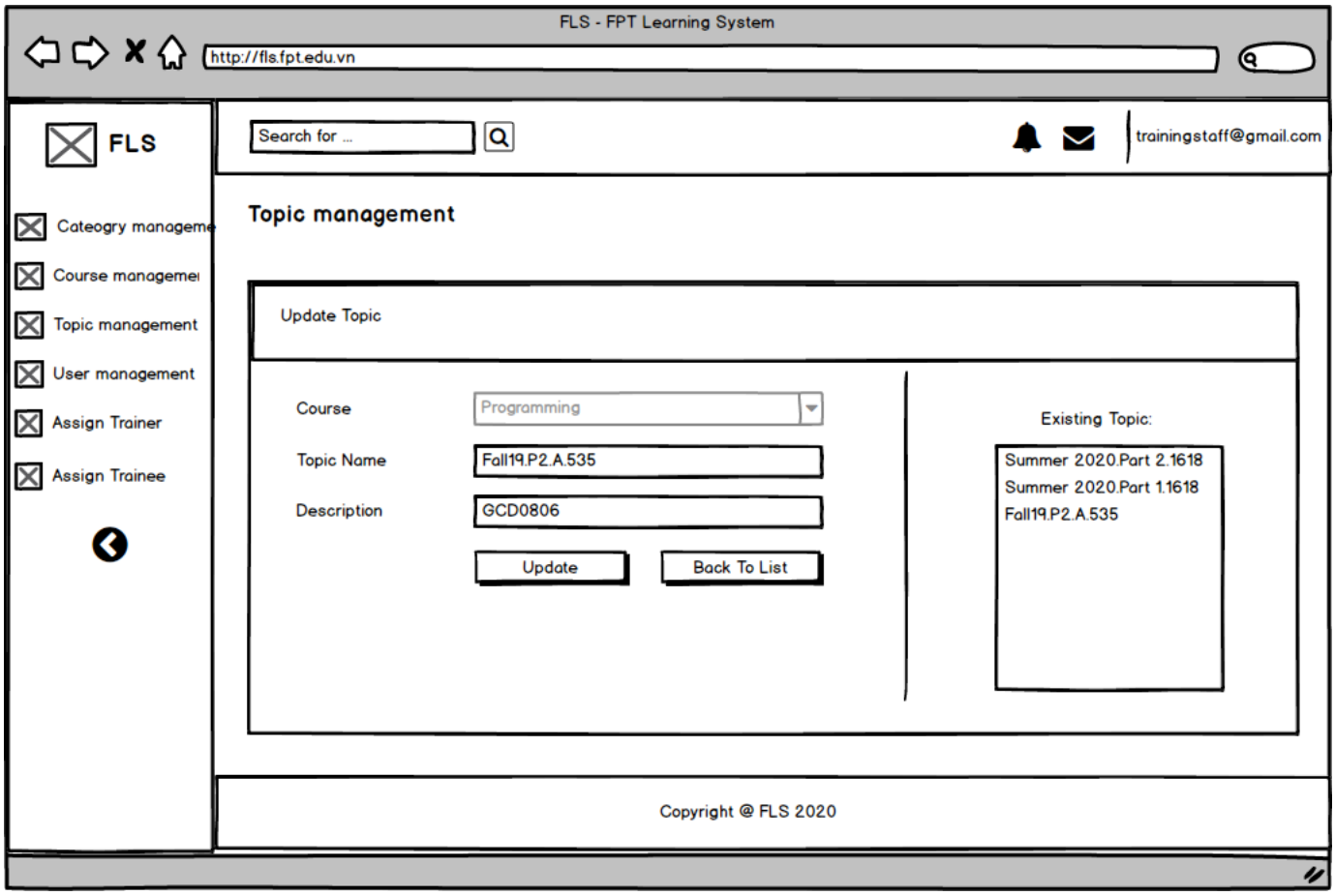
Picture 21. Training Staff - Topic management delete page

- TrainingStaff – Topic management – Create



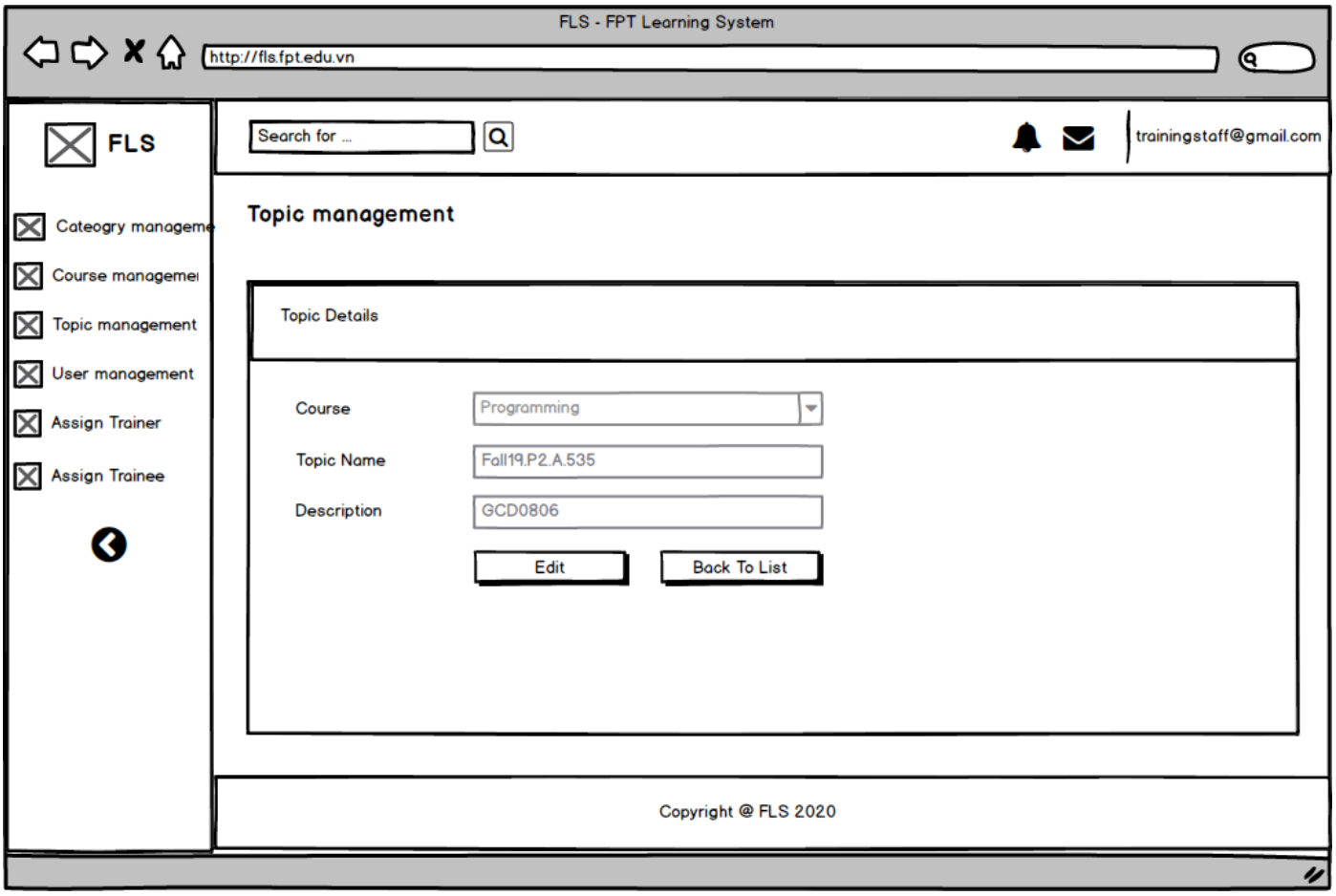
Picture 22. Training Staff - Topic management create page

- TrainingStaff – Topic management – Edit



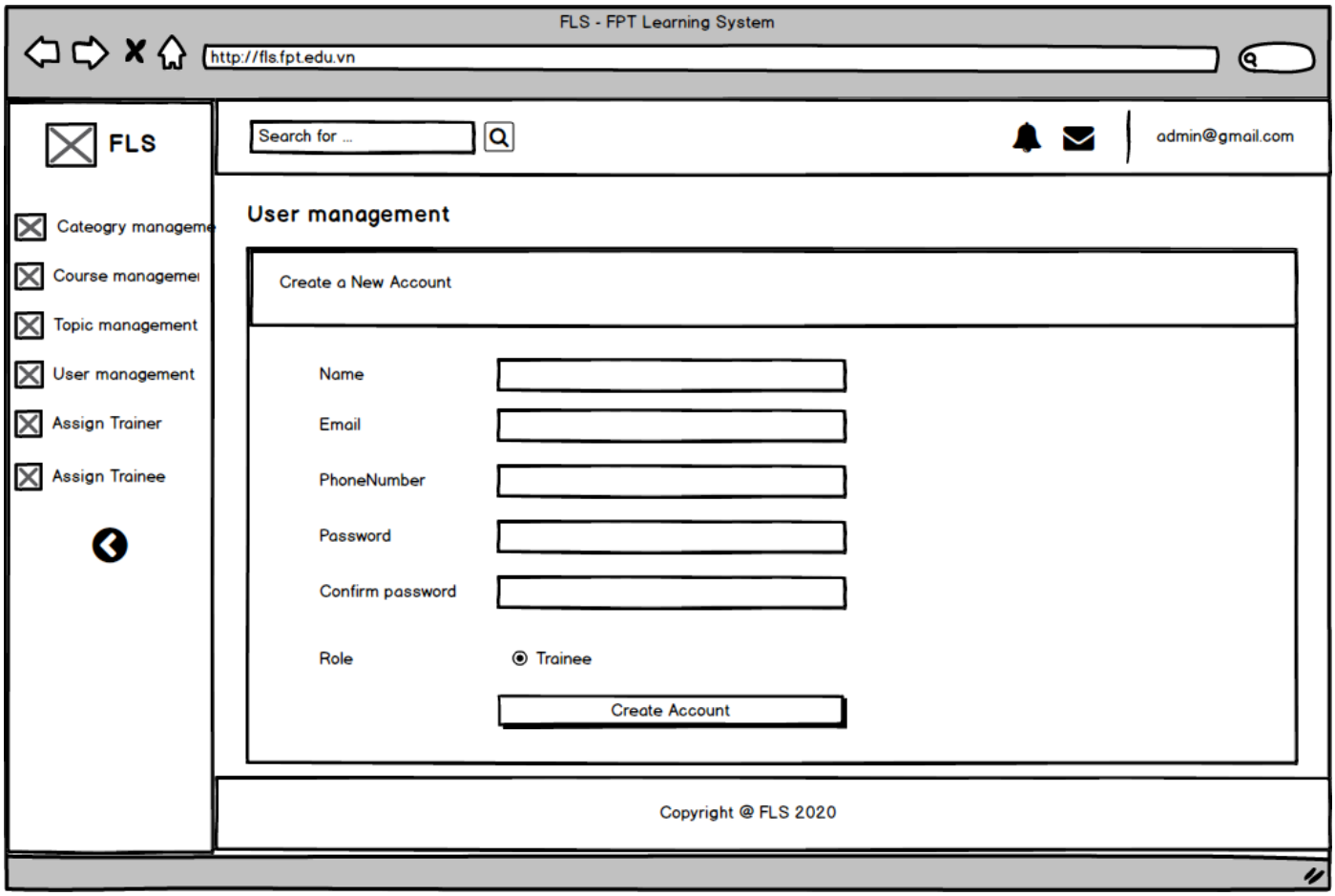
Picture 23. Training Staff - Topic management details edit

- TrainingStaff – Topic management – Details



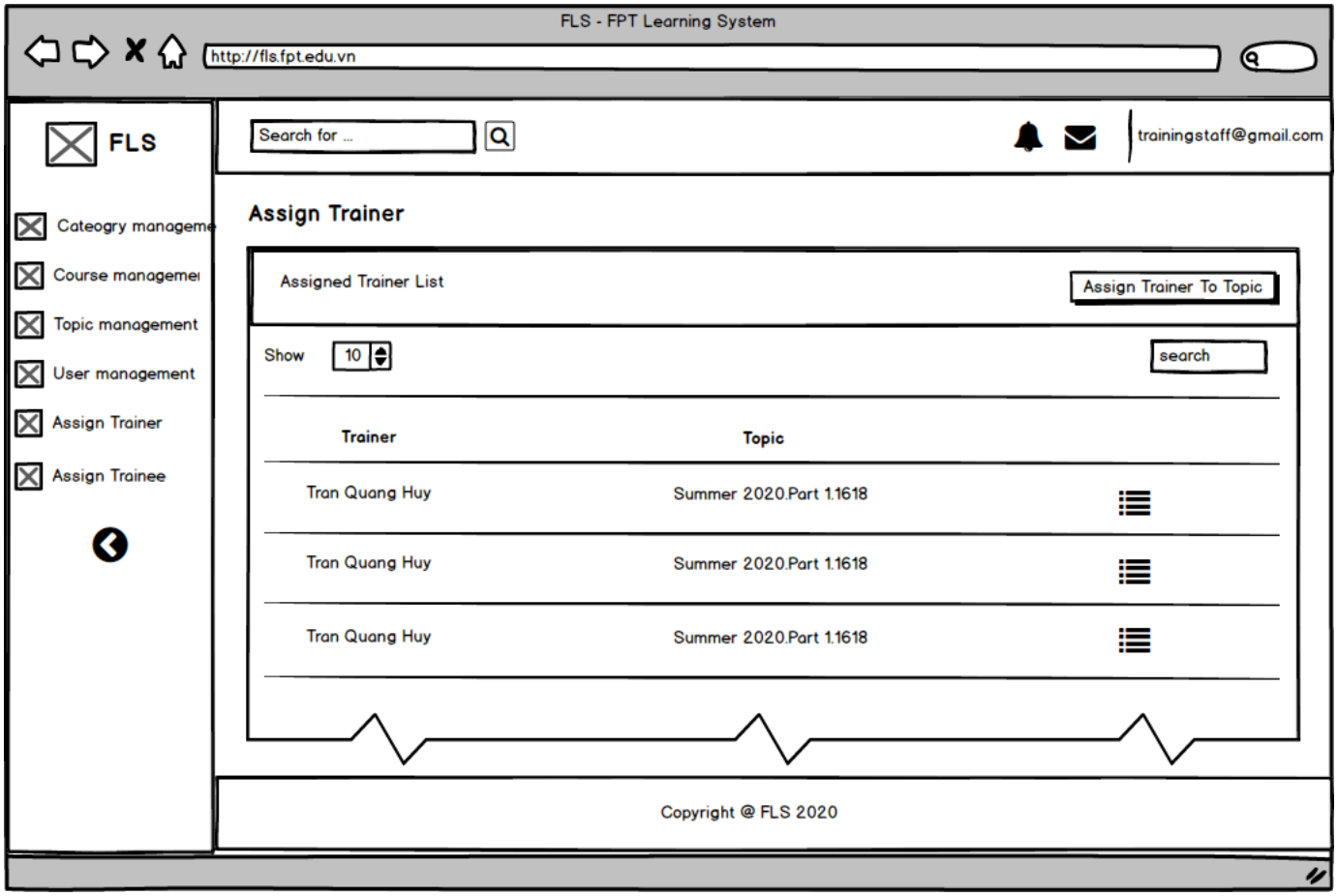
Picture 24. Training Staff - Topic management details page

- TrainingStaff – User management – Create



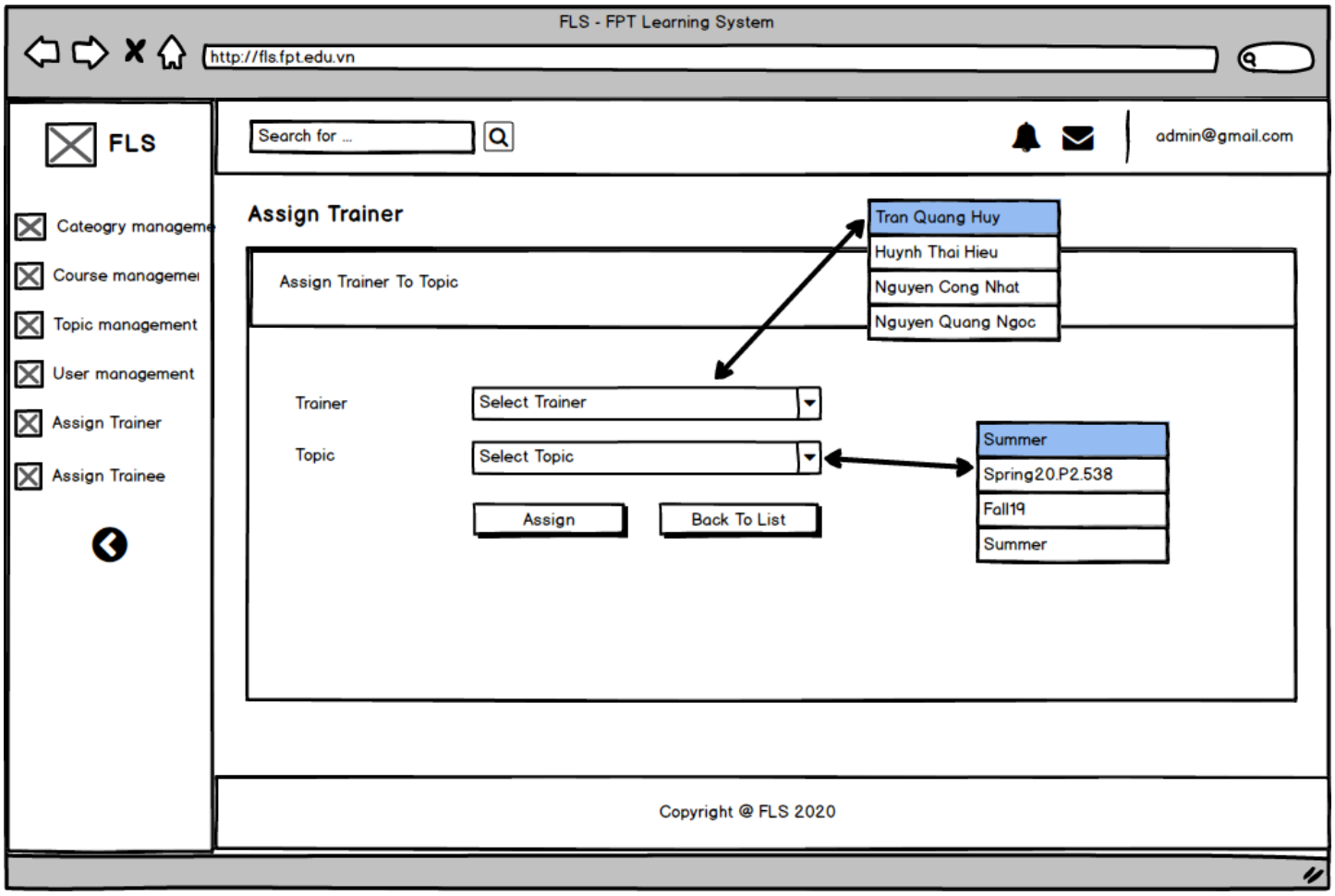
Picture 25. Training Staff - User management create page

- TrainingStaff – Assign Trainer – Index



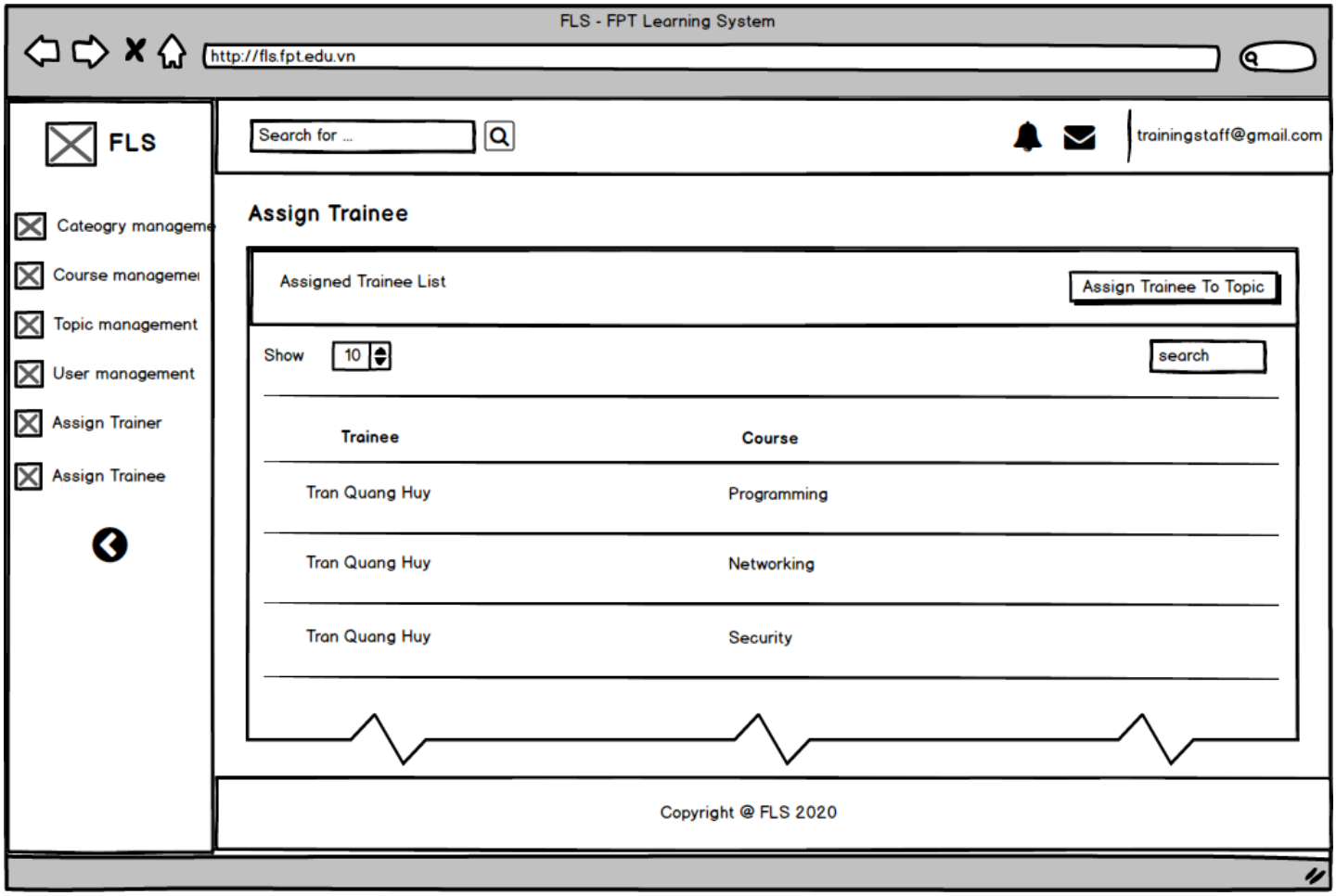
Picture 26. Training Staff - Assign Trainer index page

- TrainingStaff – Assign Trainer – Create



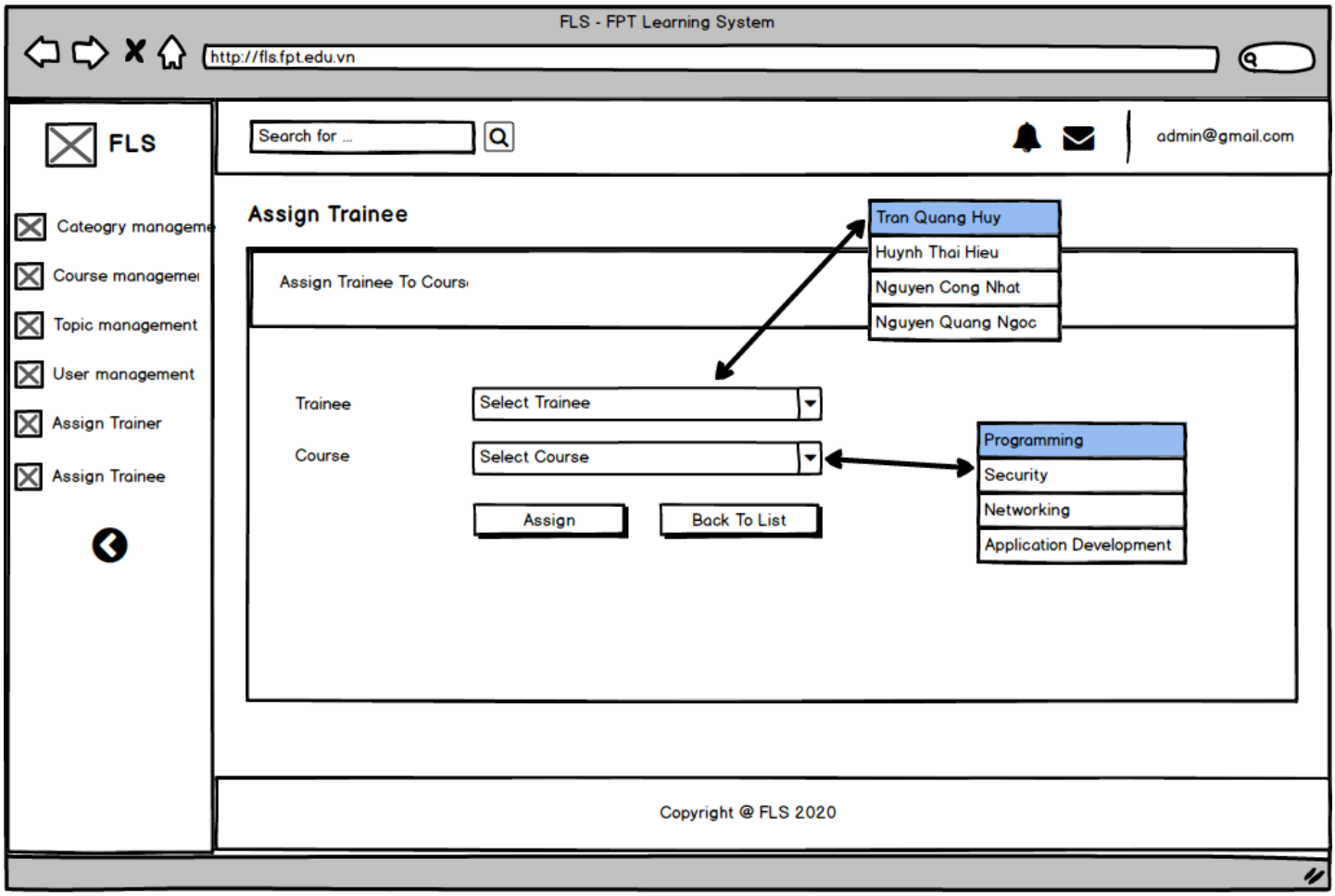
Picture 27. Training Staff - Assign Trainer create page

- TrainingStaff – Assign Trainee – Index



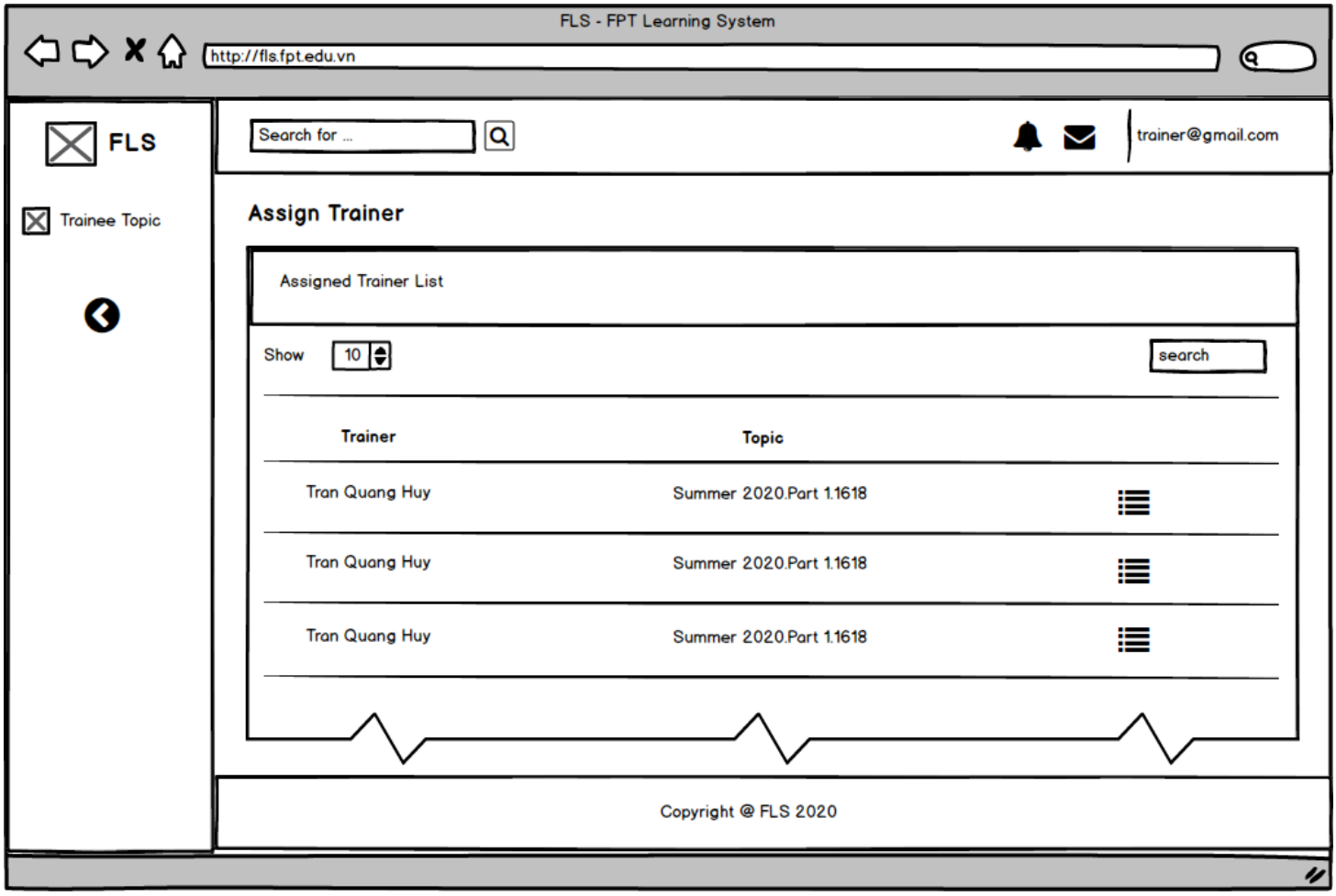
Picture 28. Training Staff - Assign Trainee index page

- TrainingStaff – Assign Trainee – Create



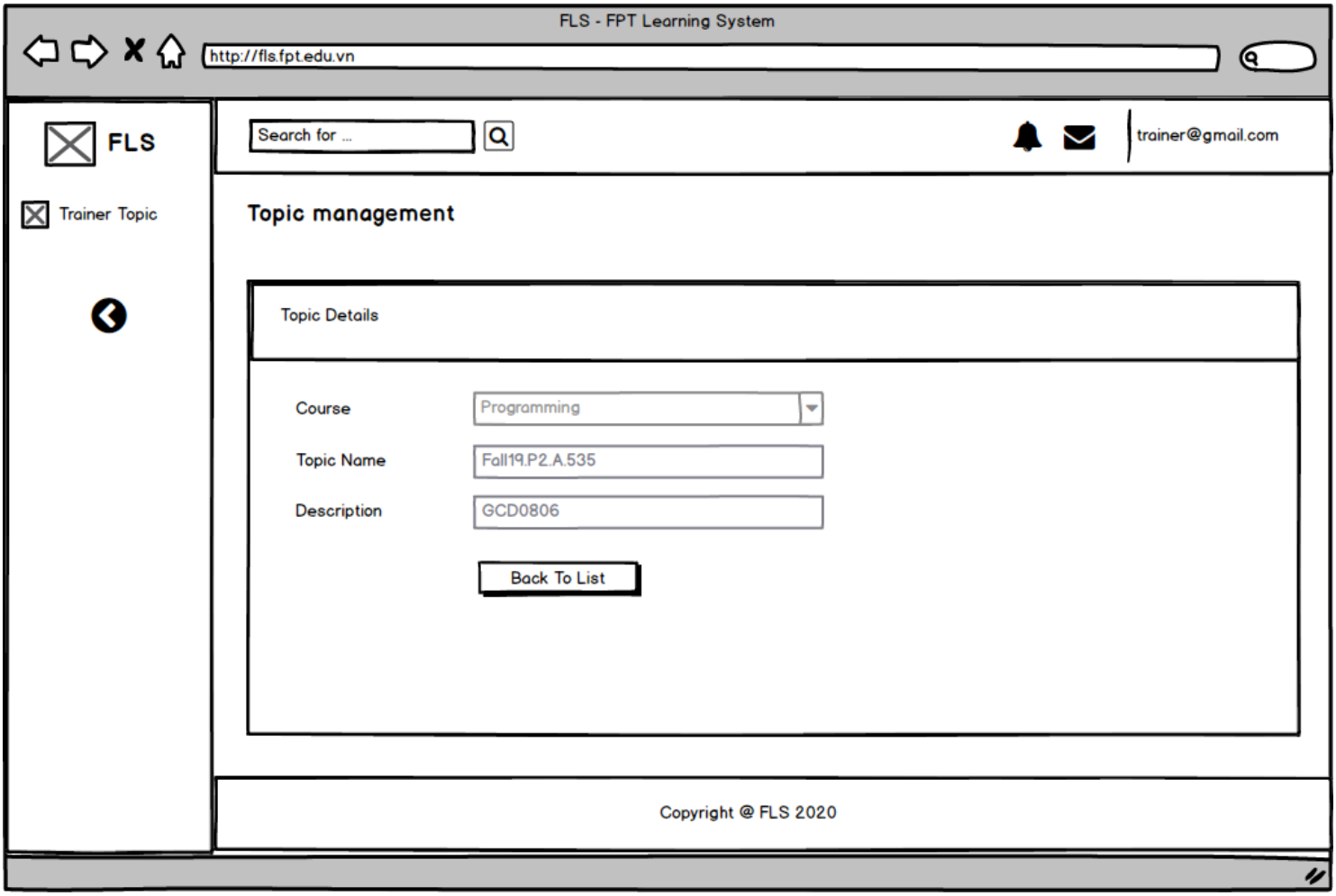
Picture 29. Training Staff - Assign Trainee create page

- Trainer – Trainer Topic – Index



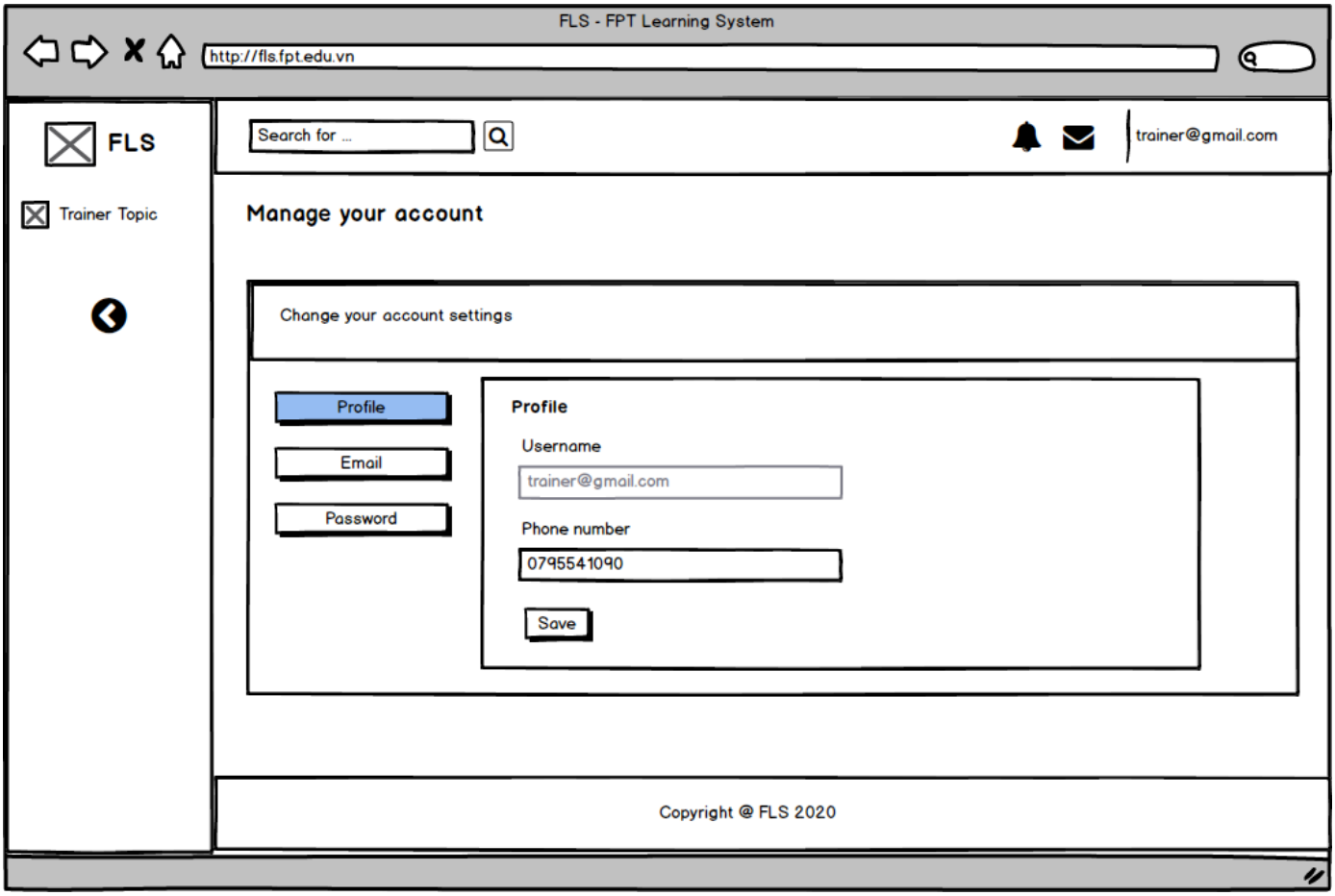
Picture 30. Trainer - Trainer Topic index page

- Trainer – Trainer Topic – Details



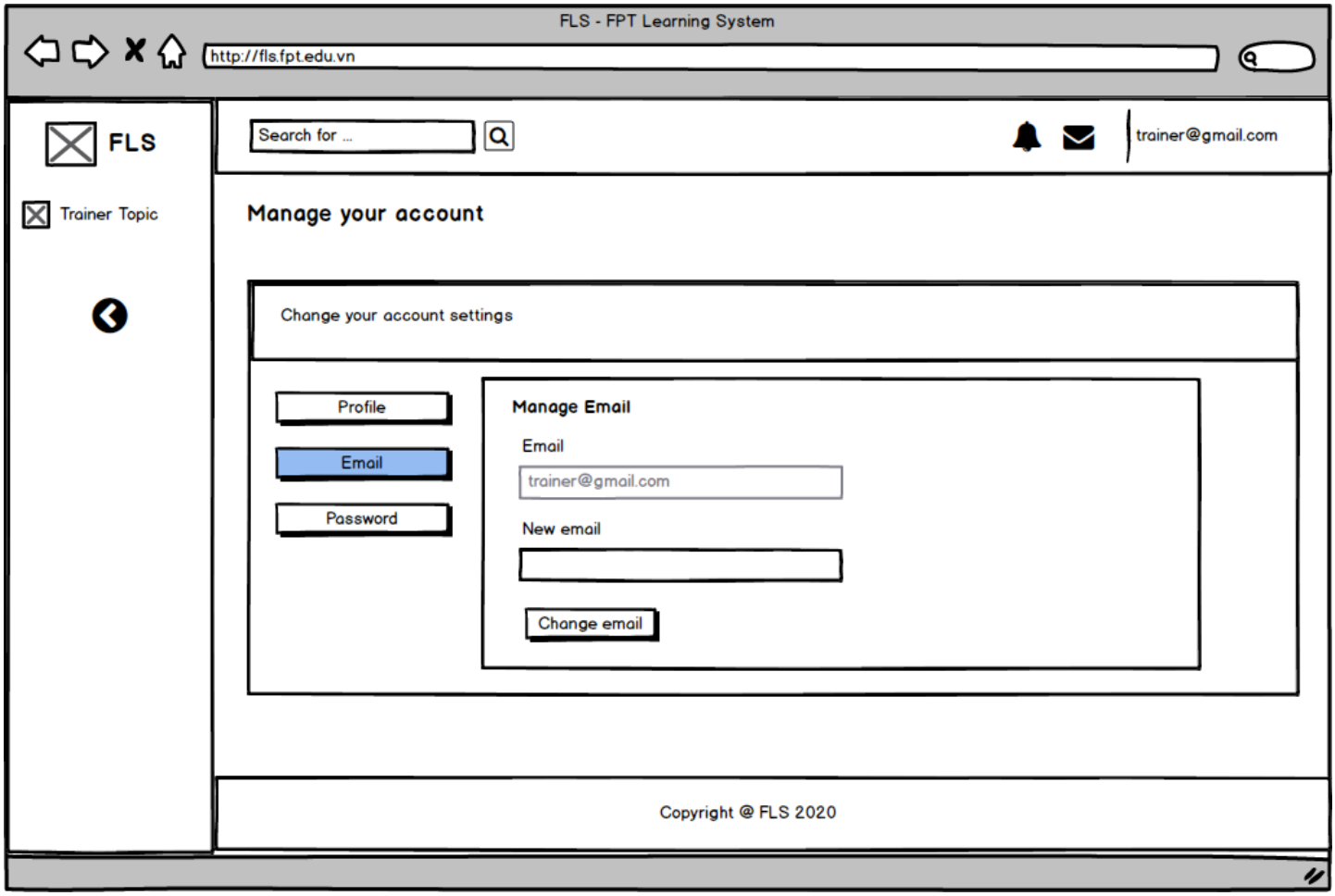
Picture 31. Trainer - Trainer topic details page

- Trainer – Account – Profile



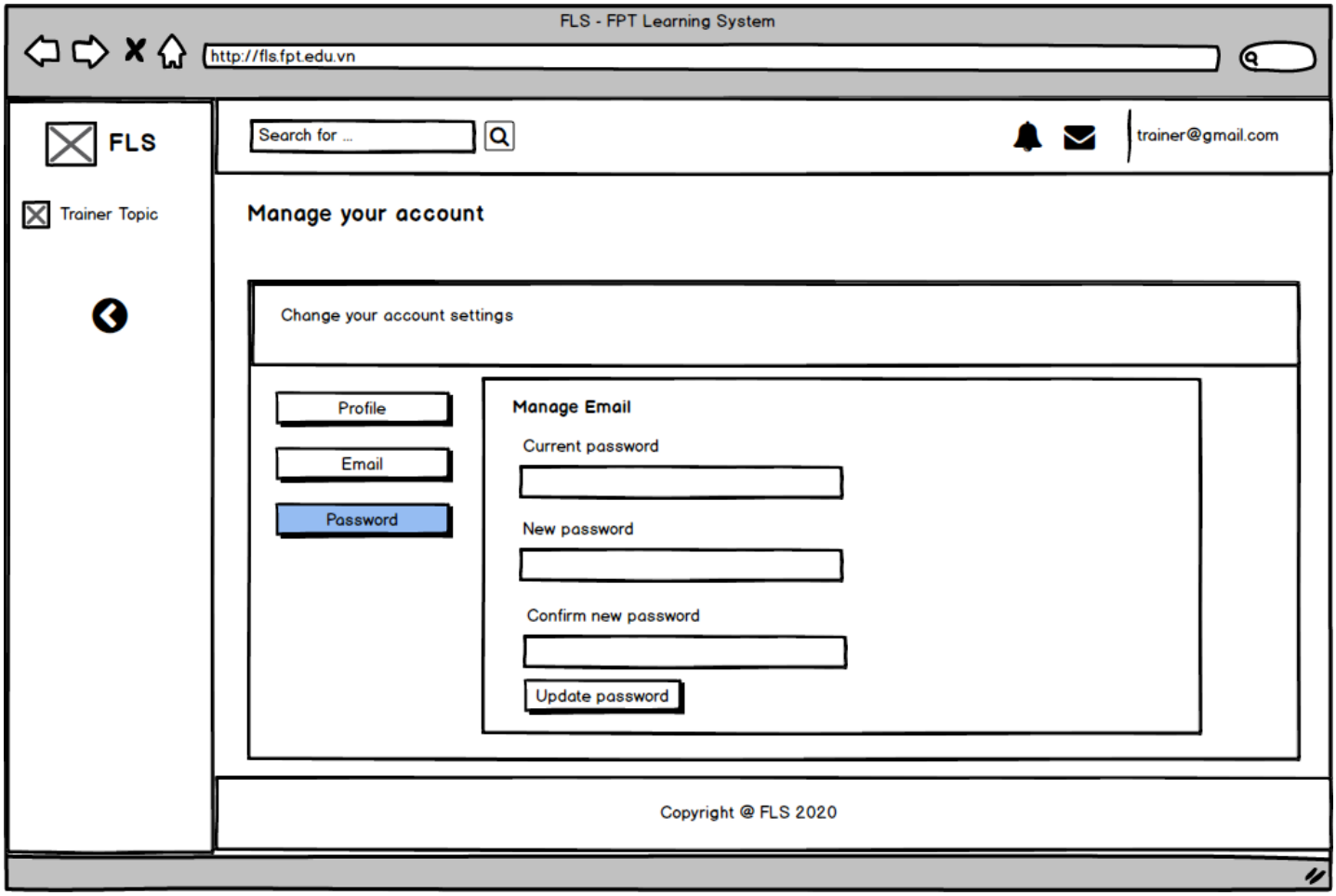
Picture 32. Trainer - Account profile page

- Trainer – Account – Email



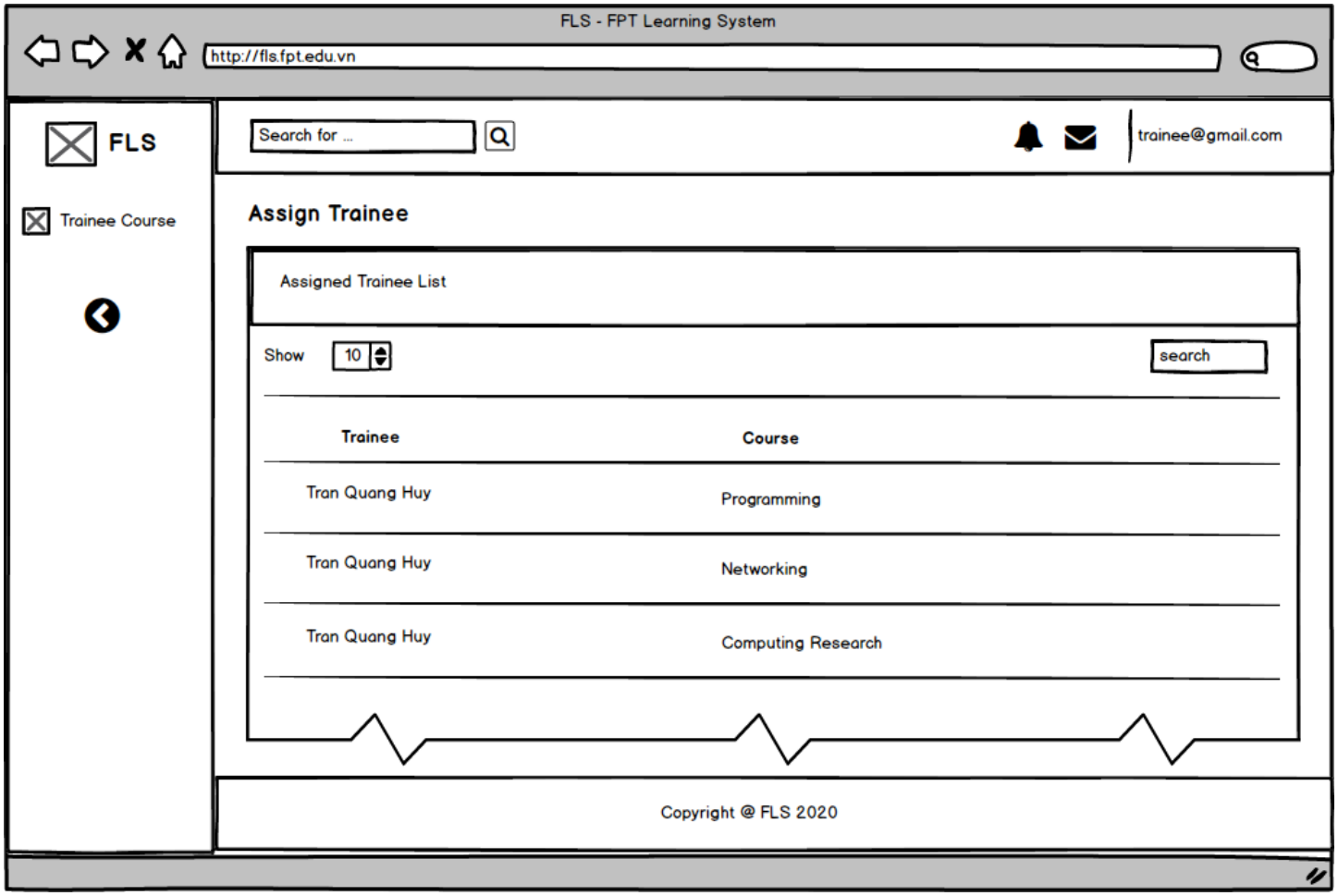
Picture 33. Trainer - Account email page

- Trainer – Account – Password



Picture 34. Trainer - Account password page

- Trainee – Trainee Course – Index



Picture 35. Trainee - Trainee Course index page

3.2 Functional Requirements

3.2.1 Administrator login

Use case Description:

Name	Administrator login	**ID**	UC_AD01
Description	Administrator login into FLS through the login page		
Actor	Administrator	**Trigger**	Click on the “login” button
Pre-condition	The login page loads completely		
Post-condition	Account inputted are valid and return to default page with the admin role		

Activities:

Actor		**System**	
Main flow: Login Successfully			
1	Administrator input the username/password and clicks the Login button on the login page		-
	-	2	Check the validation of Username and Password
	-	3	Check if the role is Administrator or Not
	-	4	Move to the default page with the admin role

Business Rules:

Rule No.	Rule	Description

System message:

Message No.	Message	Case
1	The Username/Password is incorrect. Please check again!	Wrong Username/ Password
2	The Username/Password does not exist.	The account doesn’t exist in the database
3	Login successfully!	Correct Username & Password

3.2.2 Administrator view Administrator profile

Use case Description:

Name	Administrator view Administrator profile	**ID**	UC_AD02
Description	Administrator view administrator profile through the administrator page		
Actor	Administrator	**Trigger**	Click on the "Administrator profile" button
Pre-condition	The administrator login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: View Successfully			
1	Click on the "Administrator profile" button		-
	-	2	The system move to the administrator profile page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
-	-	-

3.2.3 Administrator view Training staff profile

Use case Description:

Name	Administrator view Training staff profile	**ID**	UC_AD03
Description	Administrator view Training staff profile through the administrator page		
Actor	Administrator	**Trigger**	Click on the "User management" button
Pre-condition	The administrator login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: View Successfully			
1	Click on the "User management" button		-
	-	2	The system move to the Training staff profile page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
-	-	-

3.2.4 Administrator view Trainer account

Use case Description:

Name	Administrator view Trainer profile	**ID**	UC_AD04
Description	Administrator view Trainer profile through the administrator page		
Actor	Administrator	**Trigger**	Click on the "User management" button
Pre-condition	The administrator login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: View Successfully			
1	Click on the "User management" button		-
	-	2	The system move to the Trainer profile page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
-	-	-

3.2.5 Administrator view Trainee profile

Use case Description:

Name	Administrator view Trainee profile	**ID**	UC_AD05
Description	Administrator view trainee profile through the administrator page		
Actor	Administrator	**Trigger**	Click on the "User management" button
Pre-condition	The administrator login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: View Successfully			
1	Click on the "User management" button		-
	-	2	The system move to the Trainee profile page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
-	-	-

3.2.6 The administrator creates a new user account for Training staff

Use case Description:

Name	Administrator creates Training staff account	**ID**	UC_AD06
Description	Administrator creates Training staff account through the administrator page		
Actor	Administrator	**Trigger**	Click on the "New User" button
Pre-condition	The administrator login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Create Successfully			
1	Click on the "New User" button		-
	-	2	The system moves to the "New Account" page
3	Input the Username/Password of the new user		-
4	Select role "Training staff"		-
	-	5	Check the validation of Username and Password
6	Click on the "Create" button		-
	-	7	Create new account with Username and Password assigned in database
	-	8	The system moves to the "User management" page

Business Rules:

Rule No.	Rule	Description
1	Usernames are unique instances	Each account has a unique username

System message:

Message No.	Message	Case
1	"This account has already existed!"	Username has already existed in database
2	"Invalid password, please try again!"	Invalid password
3	"Please choose the role"	Forgot to set the role
4	"Account created successfully!"	Account has been created and saved into database

3.2.7 The administrator creates a new user account for Trainer

Use case Description:

Name	Administrator creates Trainer account	**ID**	UC_AD07
Description	Administrator creates Trainer account through the administrator page		
Actor	Administrator	**Trigger**	Click on the "New User" button
Pre-condition	The administrator login successfully		
Post-condition	-		

Activities:

Actor		System	
Main flow: Create Successfully			
1	Click on the "New User" button		-
	-	2	The system moves to the "New Account" page
3	Input the Username/Password of the new user		-
4	Select role "Trainer"		-
	-	5	Check the validation of Username and Password
6	Click on the "Create" button		-
	-	7	Create new account with Username and Password assigned in database
	-	8	The system moves to the "User management" page

Business Rules:

Rule No.	Rule	Description
1	Usernames are unique instances	Each account have a unique username

System message:

Message No.	Message	Case
1	"This account has already existed! "	Username has already existed in database
2	"Invalid password, please try again!"	Invalid password
3	"Please choose the role "	Forgot to set the role
4	"Account created successfully!"	Account has been created and saved into database

3.2.8 The administrator creates a new user account for Trainee

Use case Description:

Name	Administrator creates Trainee account	**ID**	UC_AD08
Description	Administrator creates Trainee account through the administrator page		
Actor	Administrator	**Trigger**	Click on the "New User" button
Pre-condition	The administrator login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Create Successfully			
1	Click on the "New User" button		-
	-	2	The system moves to the "New Account" page
3	Input the Username/Password of the new user		-
4	Select role "Trainee"		-
	-	5	Check the validation of Username and Password
6	Click on the "Create" button		-
	-	7	Create new account with Username and Password assigned in database
	-	8	The system moves to the "User management" page

Business Rules:

Rule No.	Rule	Description
1	Usernames are unique instances	Each account has a unique username

System message:

Message No.	Message	Case
1	"This account has already existed! "	Username has already existed in database
2	"Invalid password, please try again!"	Invalid password
3	"Please choose the role "	Forgot to set the role
4	"Account created successfully!"	Account has been created and saved into database

3.2.9 Administrator edit Administrator account

Use case Description:

Name	Administrator edit Administrator account	**ID**	UC_AD09
Description	Administrator edit Administrator account through the administrator page		
Actor	Administrator	**Trigger**	Click on the "Profile" button
Pre-condition	The administrator login successfully and "Administrator dashboard" page load completely		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Edited Successfully			
1	Click on the "Profile" button		-
	-	2	The system moves to the "Profile" page with Administrator Id
3	Input the Username/Password of the new user		-
	-	4	Check the validation of Username and Password
5	Click on the "Save" button		-
	-	6	Create new account with Username and Password assigned in database
	-	7	The system moves to the "Administrator profile" page

Business Rules:

Rule No.	Rule	Description
1	Usernames are unique instances	Each account have a unique username

System message:

Message No.	Message	Case
1	"This account has already existed!"	Username has already existed in database
2	"Invalid password, please try again!"	Invalid password
3	"Please choose the role"	Forgot to set the role
4	"Account updated successfully!"	Account has been updated and saved into database

3.2.10 Administrator edit Training staff account

Use case Description:

Name	Administrator edit Training staff account	**ID**	UC_AD10
Description	Administrator edit Training staff account through the administrator page		
Actor	Administrator	**Trigger**	Click on the "Edit" button
Pre-condition	The administrator login successfully and "Administrator dashboard" page load completely		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Edited Successfully			
1	Click on the "Edit" button		-
	-	2	The system move to the "Edit" page with Administrator Id
3	Input the Username/Password of the new user		-
	-	4	Check the validation of Username and Password
5	Click on the "Update" button		-
	-	6	Create new account with Username and Password assigned in database
	-	7	The system move to the "User management" page

Business Rules:

Rule No.	Rule	Description
1	Usernames are unique instances	Each account has a unique username

System message:

Message No.	Message	Case
1	"This account has already existed!"	Username has already existed in database
2	"Invalid password, please try again!"	Invalid password
3	"Please choose the role"	Forgot to set the role
4	"Account updated successfully!"	Account has been updated and saved into database

3.2.11 Administrator edit Trainer account

Use case Description:

Name	Administrator edit Trainer account	**ID**	UC_AD11
Description	Administrator edit Trainer account through the administrator page		
Actor	Administrator	**Trigger**	Click on the "Edit" button
Pre-condition	The administrator login successfully and "Administrator dashboard" page load completely		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Edited Successfully			
1	Click on the "Edit" button		-
	-	2	The system move to the "Edit" page with Administrator Id
3	Input the Username/Password of the new user		-
	-	4	Check the validation of Username and Password
5	Click on the "Update" button		-
	-	6	Create new account with Username and Password assigned in database
	-	7	The system move to the "User management" page

Business Rules:

Rule No.	Rule	Description
1	Usernames are unique instances	Each account have a unique username

System message:

Message No.	Message	Case
1	"This account has already existed!"	Username has already existed in database
2	"Invalid password, please try again!"	Invalid password
3	"Please choose the role"	Forgot to set the role
4	"Account updated successfully!"	Account has been updated and saved into database

3.2.12 Administrator edit Trainee account

Use case Description:

Name	Administrator edit Trainee account	**ID**	UC_AD12
Description	Administrator edit Trainee account through the administrator page		
Actor	Administrator	**Trigger**	Click on the "Edit" button
Pre-condition	The administrator login successfully and "Administrator dashboard" page load completely		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Edited Successfully			
1	Click on the "Edit" button		-
	-	2	The system move to the "Edit" page with Administrator Id
3	Input the Username/Password of the new user		-
	-	4	Check the validation of Username and Password
5	Click on the "Update" button		-
	-	6	Create new account with Username and Password assigned in database
	-	7	The system move to the "User management" page

Business Rules:

Rule No.	Rule	Description
1	Usernames are unique instances	Each account has a unique username

System message:

Message No.	Message	Case
1	"This account has already existed!"	Username has already existed in database
2	"Invalid password, please try again!"	Invalid password
3	"Please choose the role"	Forgot to set the role
4	"Account updated successfully!"	Account has been updated and saved into database

3.2.13 Administrator delete a user account

Use case Description:

Name	Administrator delete exist account	**ID**	UC_AD13
Description	Administrator delete user account through the administrator page		
Actor	Administrator	**Trigger**	Click on the “Delete” button
Pre-condition	The administrator login successfully and “Administrator dashboard” page load completely		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Deleted Successfully			
1	Click on the “Delete” button		-
	-	2	The system show pop-up confirm that Administrator want to delete or not
3	Administrator click on “Ok” button		-
	-	4	The account has been deleted in database
	-	5	The system move to the “User management”- page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	“Do you want to delete?”	Administrator click on the “delete” button
2	“The account has been deleted!”	The account has been deleted in database

3.2.14 Training staff login

Use case Description:

Name	Training staff login	**ID**	UC_TS01
Description	Training staff login into FLS through the login page		
Actor	Training staff	**Trigger**	Click on the “login” button
Pre-condition	The login page loads completely		
Post-condition	Account inputted are valid and return to default page with the Training staff role		

Activities:

Actor		**System**	
Main flow: Login Successfully			
1	Training staff input the username/password and clicks the Login button on the login page		-
	-	2	Check the validation of Username and Password
	-	3	Check if the role is Administrator or Not
	-	4	Move to the default page with the training staff role

Business Rules:

Rule No.	Rule	Description

System message:

Message No.	Message	Case
1	The Username/Password is incorrect. Please check again!	Wrong Username/ Password
2	The Username/Password does not exist.	The account doesn’t exist in the database
3	Login successfully!	Correct Username & Password

3.2.15 Training staff create trainee account

Use case Description:

Name	Training staff creates Trainee account	**ID**	UC_TS02
Description	Training staff creates Trainee account through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "New user" button
Pre-condition	The Training staff login successfully		
Post-condition	-		

Activities:

Actor		System	
Main flow: Created Successfully			
1	Click on the "New user" button		-
	-	2	The system move to the "New Account"page
3	Input the Username/Password of the new user		-
4	Input the trainee name, age, date of birth, education, main programming language, TOEIC score, experience details, department, location, etc		-
	-	5	Check the validation of Username and Password
6	Click on the "Create" button		-
	-	7	Create new account with Username and Password assigned in database
	-	8	The system move to the "Training staff dashboard" page

Business Rules:

Rule No.	Rule	Description
1	Usernames are unique instances	Each account have a unique username

System message:

Message No.	Message	Case
1	"This account has already existed!"	Username has already existed in database
2	"Invalid password, please try again!"	Invalid password
3	"Account created successfully!"	Account has been created and saved into database

3.2.16 Training staff view list of trainees

Use case Description:

Name	Training staff view list of trainees	**ID**	UC_TS03
Description	Training staff view list of trainees through the training staff page		
Actor	Training staff	**Trigger**	Click on the "User management" button
Pre-condition	The Training staff login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: View Successfully			
1	Click on the "User management" button		-
	-	2	The system move to the "User management" page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
-	-	-

3.2.17 Training staff view trainee profile

Use case Description:

Name	Training staff view trainee profile	**ID**	UC_TS04
Description	Training staff view list of trainees through the training staff page		
Actor	Training staff	**Trigger**	Click on the “Trainee profile” button
Pre-condition	The Training staff login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: View Successfully			
1	Click on the “Trainee profile” button		-
	-	2	The system move to the trainee profile page with trainee specific id

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
-	-	-

3.2.18 Training staff update trainee profile

Use case Description:

Name	Training staff update trainee profile	**ID**	UC_TS05
Description	Training staff update trainee profile through the training staff page		
Actor	Training staff	**Trigger**	Click on the "Edit" button
Pre-condition	The Training staff login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Updated Successfully			
1	Click on the "Edit" button		-
	-	2	The system move to the "edit trainee profile" page with trainee specific id
3	Training staff input the trainee information		
4	Click on the "Update" button		
		5	The trainee information has been updated into database

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	"Missing some information. Please fill in again!"	Trainning staff forgot to fill in some information
2	"Updated successfully!"	The trainee information has been updated into database

3.2.19 Training staff delete trainee account

Use case Description:

Name	Training staff delete exist account	**ID**	UC_TS06
Description	Training staff delete trainee account through the Training staff page		
Actor	Training staff	**Trigger**	Click on the “Delete” button
Pre-condition	The Training staff login successfully and “Trainee” page load completely		
Post-condition	-		

Activities:

	Actor		System
Main flow: Deleted Successfully			
1	Click on the “Delete” button		-
	-	2	The system show pop-up confirm that want to delete or not
3	Click on “Ok” button		-
	-	4	The account has been deleted in database
	-	5	The system move to the “User management” page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	“Do you want to delete?”	Administrator click on the “delete” button
2	“The account has been deleted!”	The account has been deleted in database

3.2.20 Training staff view list of courses

Use case Description:

Name	Training staff view list of courses	**ID**	UC_TS07
Description	Training staff view list of course through the Training staff page		
Actor	Training staff	**Trigger**	Click on the “Course management” button
Pre-condition	The Training staff login successfully and “Course” page load completely		
Post-condition	-		

Activities:

Actor		System	
Main flow: View Successfully			
1	Click on the “Course management” button		-
	-	2	The system move to the “Course management” page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case

3.2.21 Training staff add a course

Use case Description:

Name	Training staff add course	**ID**	UC_TS08
Description	Training staff add course through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "Add Course" button
Pre-condition	The Training staff login successfully and "Course" page load completely		
Post-condition	-		

Activities:

Actor		System	
Main flow: Added Successfully			
1	Click on the "Add Course" button		-
	-	2	The system move to the "Add Course" page
3	Input information about course such as course name and course description		
4	Click on the "Add" button		
		5	Check the exist Course in Database
		6	The course has been created into database
		7	Show pop-up add successful message

Business Rules:

Rule No.	Rule	Description
1	Course name is unique	Each course have a unique name

System message:

Message No.	Message	Case
1	"Course name cannot be empty"	Training staff forgot to fill course name
2	"Course created!"	Created course into database

3.2.22 Training staff update course

Use case Description:

Name	Training staff update course	**ID**	UC_TS09
Description	Training staff update course through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "Edit" button
Pre-condition	The Training staff login successfully and "Course" page load completely		
Post-condition	-		

Activities:

Actor		System	
Main flow: Updated Successfully			
1	Click on the "Edit" button		-
	-	2	The system move to the "Edit Course" page
3	Input information about course such as course name and course description		
4	Click on the "Update" button		
		5	Check the exist Course in Database
		6	The course has been updated into database
		7	Show pop-up add successful message

Business Rules:

Rule No.	Rule	Description
1	Course name is unique	Each course have a unique name

System message:

Message No.	Message	Case
1	"Course name cannot be empty"	Training staff forgot to fill course name
2	"Course updated!"	Updated course into database

3.2.23 Training staff delete course

Use case Description:

Name	Training staff delete course	**ID**	UC_TS10
Description	Training staff delete course through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "Delete Course" button
Pre-condition	The Training staff login successfully and "Course" page load completely		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Deleted Successfully			
1	Click on the "Delete Course" button		-
	-	2	The system show pop-up confirm that want to delete or not
3	Click on "Ok" button		-
	-	4	The Course has been deleted in database

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	"Do you want to delete this course?"	Administrator click on the "delete" button
2	"The course has been deleted!"	The course has been deleted in database

3.2.24 Training staff add topic into course

Use case Description:

Name	Training staff Add topic into course	**ID**	UC_TS11
Description	Training staff add topic into course through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "Add topic" button
Pre-condition	The Training staff login successfully and "Course" page load completely		
Post-condition	-		

Activities:

Actor		System	
Main flow: Added Successfully			
1	Click on the "Add topic" button		-
	-	2	The system move to the "Add topic" page
3	Input information about course such as course name and course description		-
4	Select course name		-
5	Click on the "Add" button		
		6	The topic assigned to course and create in database
		7	The system show pop-up success message

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	"Assigned topic to course successfully"	Add topic to course

3.2.25 Training staff add category

Use case Description:

Name	Training staff add category	**ID**	UC_TS12
Description	Training staff add category through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "Add Category" button
Pre-condition	The Training staff login successfully and "Category" page load completely		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Added Successfully			
1	Click on the "Add Category" button		-
	-	2	The system move to the "Add Category" page
3	Input information about category such as category name and course description		
4	Click on the "Add" button		
		5	Check the exist Category in Database
		6	The Category has been created into database
		7	Show pop-up add successful message

Business Rules:

Rule No.	Rule	Description
1	Category name is unique	Each Category have a unique name

System message:

Message No.	Message	Case
1	"Category name cannot be empty"	Training staff forgot to fill Category name
2	"Category created!"	Created Caregory into database

3.2.26 Training staff update category

Use case Description:

Name	Training staff update category	**ID**	UC_TS13
Description	Training staff update category through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "Edit Category" button
Pre-condition	The Training staff login successfully and "Category" page load completely		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Updated Successfully			
1	Click on the "Edit Category" button		-
	-	2	The system move to the "Edit Category" page
3	Input information about course such as course name and course description		
4	Click on the "Update" button		
		5	Check the exist Category in Database
		6	The Category has been updated into database
		7	Show pop-up add successful message

Business Rules:

Rule No.	Rule	Description
1	Category name is unique	Each Category have a unique name

System message:

Message No.	Message	Case
1	"Category name cannot be empty"	Training staff forgot to fill Category name
2	"Category updated!"	Updated Category into database

3.2.27 Training staff delete category

Use case Description:

Name	Training staff delete Category	**ID**	UC_TS14
Description	Training staff delete Category through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "Delete Category" button
Pre-condition	The Training staff login successfully and "Category" page load completely		
Post-condition	-		

Activities:

Actor		System	
Main flow: Deleted Successfully			
1	Click on the "Delete Category" button		-
	-	2	The system show pop-up confirm that want to delete or not
3	Click on "Yes" button		-
	-	4	The Category has been deleted in database

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	"Do you want to delete this Category?"	Administrator click on the "delete" button
2	"The Category has been deleted!"	The Category has been deleted in database

3.2.28 Training staff add course into a category

Use case Description:

Name	Training staff add course to category	**ID**	UC_TS15
Description	Training staff add course to category through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "Course to category" button
Pre-condition	The Training staff login successfully and "Course" page load completely		
Post-condition	-		

Activities:

Actor		System	
Main flow: Added Successfully			
1	Click on the "Add topic" button		-
	-	2	The system move to the "Add topic" page
3	Input information about course such as course name and course description		-
4	Select course name		-
5	Click on the "Add" button		
		6	The topic assigned to course and create in database
		7	The system show pop-up success message

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	"Assigned topic to course successfully"	Add topic to course

3.2.29 Training staff view trainer profile

Use case Description:

Name	Training staff view trainer profile	**ID**	UC_TS16
Description	Training staff view trainer profile through the training staff page		
Actor	Training staff	**Trigger**	Click on the “User management” button
Pre-condition	The Training staff login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: View Successfully			
1	Click on the “Trainer profile” button		-
	-	2	The system move to the Trainer profile page with trainee specific id

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
-	-	-

3.2.30 Training staff add trainer profile

Use case Description:

Name	Training staff creates Trainer account	**ID**	UC_TS17
Description	Training staff creates Trainer account through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "New User" button
Pre-condition	The Training staff login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Created Successfully			
1	Click on the "New User" button		-
	-	2	The system move to the "New Account"page
3	Input the Username/Password of the new user		-
4	Input the Trainer name, external or internal type, working place, telephone, and email address.		-
	-	5	Check the validation of Username and Password
6	Click on the "Create" button		-
	-	7	Create new account with Username and Password assigned in database
	-	8	The system move to the "User management" page

Business Rules:

Rule No.	Rule	Description
1	Usernames are unique instances	Each account have a unique username

System message:

Message No.	Message	Case
1	"This account has already existed!"	Username has already existed in database
2	"Invalid password, please try again!"	Invalid password
3	"Account created successfully!"	Account has been created and saved into database

3.2.31 Training staff update trainer profile

Use case Description:

Name	Training staff update trainer profile	**ID**	UC_TS18
Description	Training staff update trainer profile through the training staff page		
Actor	Training staff	**Trigger**	Click on the "Edit" button
Pre-condition	The Training staff login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Updated Successfully			
1	Click on the "Edit" button		-
	-	2	The system move to the "edit trainer profile" page with trainee specific id
3	Training staff input the trainer information		
4	Click on the "Update" button		
		5	The trainer information has been updated into database

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	"Missing some information. Please fill in again!"	Trainning staff forgot to fill in some information
2	"Updated successfully!"	The trainer information has been updated into database

3.2.32 Training staff delete trainer profile

Use case Description:

Name	Training staff delete exist trainer	**ID**	UC_TS19
Description	Training staff delete trainer account through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "Delete" button
Pre-condition	The Training staff login successfully and "Trainer" page load completely		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Deleted Successfully			
1	Click on the "Delete" button		-
	-	2	The system show pop-up confirm that want to delete or not
3	Click on "Ok" button		-
	-	4	The account has been deleted in database
	-	5	The system move to the "Trainer" page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	"Do you want to delete this account?"	Administrator click on the "delete" button
2	"The account has been deleted!"	The account has been deleted in database

3.2.33 Training staff assign a trainer to topic

Use case Description:

Name	Training staff assign trainer to topic	**ID**	UC_TS20
Description	Training staff asisign trainer to topic through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "Assign trainer" button
Pre-condition	The Training staff login successfully		
Post-condition	Redirect to "Assign Trainer" page		

Activities:

Actor		**System**	
Main flow: Asign trainer Successfully			
1	Click on the "Assign trainer" button		-
2	Select topic from dropdown list of topic		-
	-	3	The system show pop-up confirm that want to assign or not
4	Click on "Yes" button		-
	-	5	The system update "trainer" into topic in database
	-	6	The system move to the "Assign Trainer" page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	"Do you want to replace this trainer?"	Confirmation when another trainer is already assigned to topic
2	"Do you want to assign this trainer to this topic?"	Confirmation when assign Trainer to topic
3	"The trainer has been assigned to topic!"	The trainer has been assigned to topic in database

3.2.34 Training staff assign trainee to course

Use case Description:

Name	Training staff assign trainee to course	**ID**	UC_TS21
Description	Training staff asisign trainee to course through the Training staff page		
Actor	Training staff	**Trigger**	Click on the "Assign trainee" button
Pre-condition	The Training staff login successfully		
Post-condition	Redirect to "Assign Trainee" page		

Activities:

Actor		**System**	
Main flow: Asign trainee Successfully			
1	Click on the "Assign trainee" button		-
2	Select topic from dropdown list of topic		-
	-	3	The system show pop-up confirm that want to assign trainee or not
4	Click on "Yes" button		-
	-	5	The system update "trainee" into course in database
	-	6	The system move to the "Assign trainee" page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	"Do you want to assign this trainer to this topic?"	Confirmation when assign Trainer to topic
2	"The trainer has been assigned to topic!"	The trainer has been assigned to topic in database

3.2.35 Trainer login

Use case Description:

Name	Trainer login	**ID**	UC_TN01
Description	Trainer login into FLS through the login page		
Actor	Trainer	**Trigger**	Click on the “login” button
Pre-condition	The login page loads completely		
Post-condition	Account inputted are valid and return to default page with the Trainer role		

Activities:

Actor		**System**	
Main flow: Login Successfully			
1	Trainer input the username/password and clicks the Login button on the login page		-
	-	2	Check the validation of Username and Password
	-	3	Check if the role is Administrator or Not
	-	4	Move to the default page with the Trainer role

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	The Username/Password is incorrect. Please check again!	Wrong Username/ Password
2	The Username/Password does not exist.	The account doesn’t exist in the database
3	Login successfully!	Correct Username & Password

3.2.36 Trainer update current trainer profile

Use case Description:

Name	Trainer update current Trainer profile	**ID**	UC_TN02
Description	Trainer update current Trainer profile through the Trainer page		
Actor	Trainer	**Trigger**	Click on the "Update profile" button
Pre-condition	The Trainer login successfully		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: Updated Successfully			
1	Click on the "Update profile" button		-
	-	2	The system move to the "edit Trainer profile" page with trainee specific id
3	Trainer input the information such as Trainer name, External or Internal Type, education, working place, telephone, and email address.		
4	Click on the "Update" button		
		5	The Trainer information has been updated into database

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	"Missing some information. Please fill in again!"	Trainer forgot to fill in some information
2	"Updated successfully!"	The Trainer information has been updated into database

3.2.37 Trainer view list of topic

Use case Description:

Name	Trainer view list of topic	**ID**	UC_TN03
Description	Training staff view list of topic through the Trainer page		
Actor	Trainer	**Trigger**	Click on the “Topic Details” button
Pre-condition	The Trainer login successfully, “Topic” page load completely and the trainer has been assigned to topic		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: View Successfully			
1	Click on the “Topic Details” button		-
	-	2	The system move to the “Topic Details” page

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case

3.2.38 Trainee login

Use case Description:

Name	Trainee login	**ID**	UC_TE01
Description	Trainee login into FLS through the login page		
Actor	Trainee	**Trigger**	Click on the “login” button
Pre-condition	The login page loads completely		
Post-condition	Account inputted are valid and return to default page with the Trainee role		

Activities:

Actor		**System**	
Main flow: Login Successfully			
1	Trainee input the username/password and clicks the Login button on the login page		-
	-	2	Check the validation of Username and Password
	-	3	Check if the role is Administrator or Not
	-	4	Move to the default page with the Trainee role

Business Rules:

Rule No.	Rule	Description
-	-	-

System message:

Message No.	Message	Case
1	The Username/Password is incorrect. Please check again!	Wrong Username/ Password
2	The Username/Password does not exist.	The account doesn’t exist in the database
3	Login successfully!	Correct Username & Password

3.2.39 Trainee view list of courses

Use case Description:

Name	Trainee view list of courses	**ID**	UC_TE02
Description	Trainee view list of course through the Trainee page		
Actor	Trainee	**Trigger**	Click on the "Trainee Course" button
Pre-condition	The Trainee login successfully, "Course" page load completely and the Trainee has been assigned to topic		
Post-condition	-		

Activities:

Actor		**System**	
Main flow: View Successfully			
1	Click on the "Trainee Course" button		-
	-	2	The system move to the "Trainee Course" page

Business Rules:

Rule No.	**Rule**	**Description**
-	-	-

System message:

Message No.	**Message**	**Case**

3.3 Use Case Model

3.3.1 Administrator use case diagram

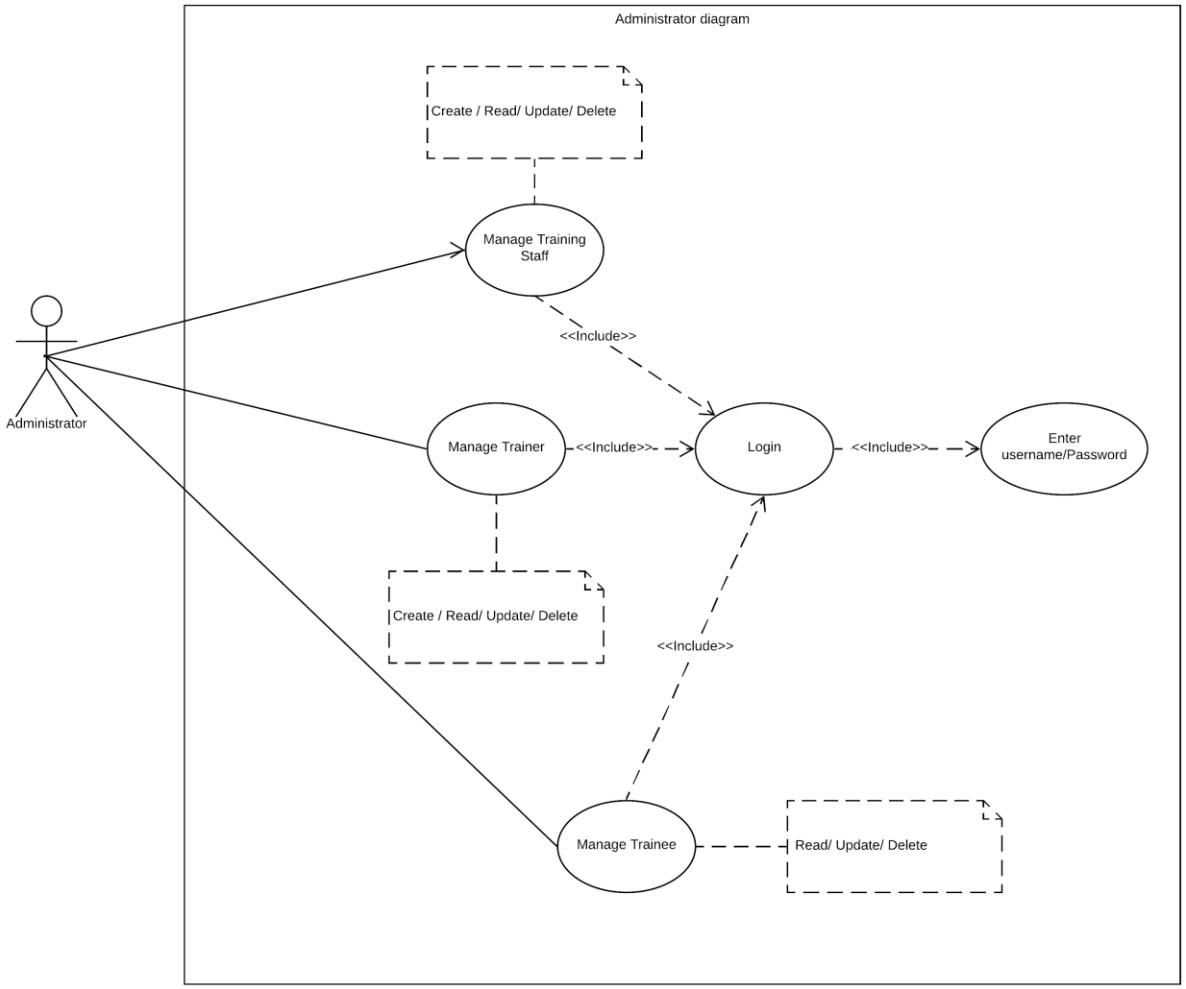


Figure 3. Administrator use case diagram

- **Purpose**: The interaction between Administrator and system.
- **Requirements Traceability:** Base on system requirement that system can used to manage trainee accounts, manage trainers, manage course categories, manage courses, manage topics, assign topic to course, assign trainer to topic, assign trainee to course and the Administrator role can login to the system, create, edit and delete new user account
- **Priority**: High
- **Actors**: Administrator
- **Extends:**
    - View/Create/Update/Delete Training staff account
    - View/Create/Update/Delete Trainer account
    - View/Create/Update/Delete Course
    - View/Create/Update/Delete Topic
    - View/Create/Update/Delete Category
- **Includes**:
    - Verify username/password

3.3.2 Training staff use case diagram

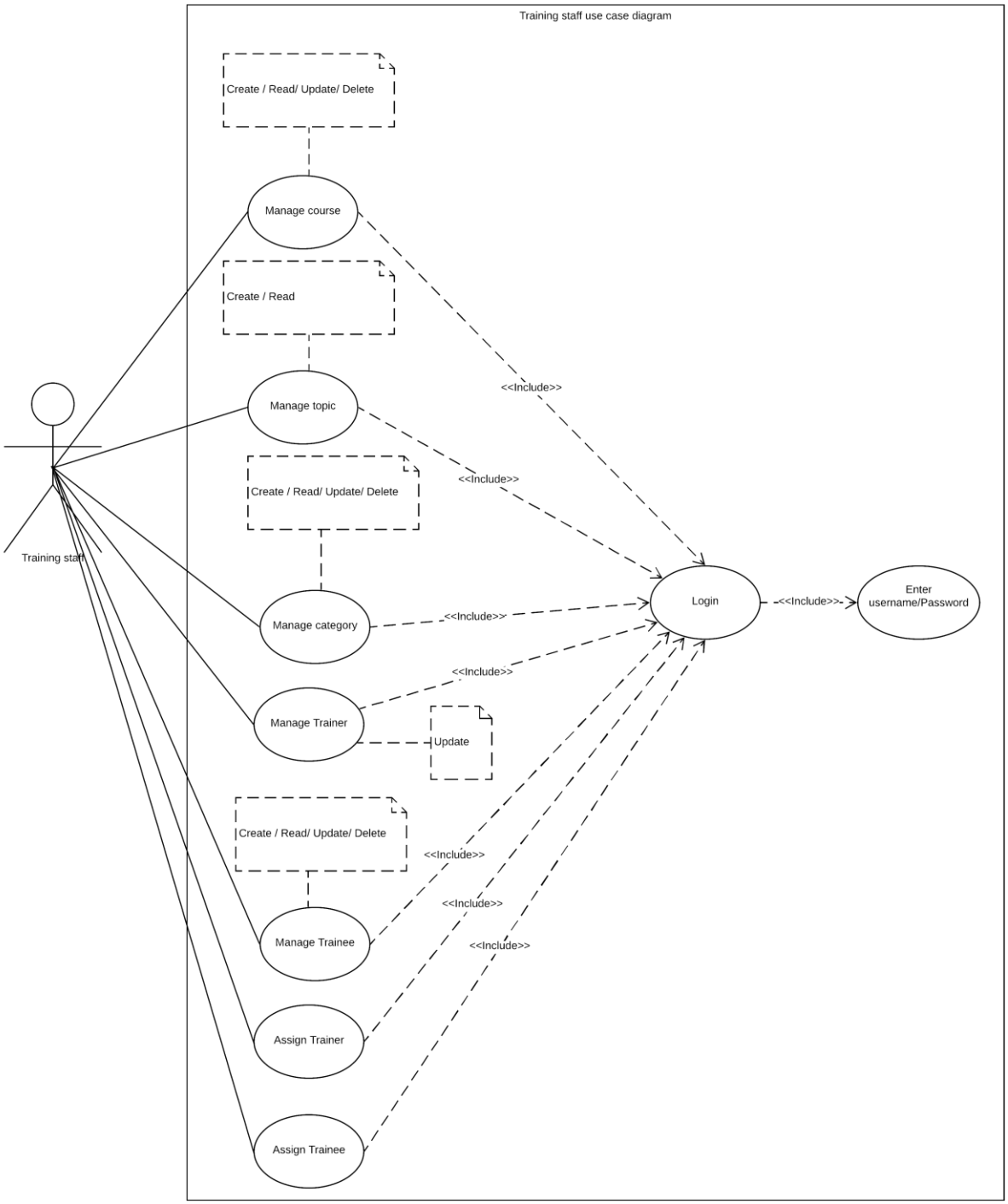


Figure 4. Training Staff use case diagram Manage contents

Purpose: The interaction between Training Staff and system.

- **Requirements Traceability:** Base on system requirement that system can used to manage trainee accounts, manage trainers, manage course categories, manage courses, manage topics, assign topic to course, assign trainer to topic, assign trainee to course and the Administrator role can login to the system, create, edit and delete new Trainer and trainee account
- **Priority**: High
- **Actors**: Training Staff
- **Extends:**
    - View /Update Training staff account
    - View/Create/Update/Delete Trainer account
    - View/Create/Update/Delete Trainee account
    - View/Create/Update/Delete Course
    - View/Create Topic
    - View/Create/Update/Delete Category
    - Assign Trainer
    - Assign Trainee
- **Includes**:
    - Verify username/password

3.3.3 Trainer use case diagram

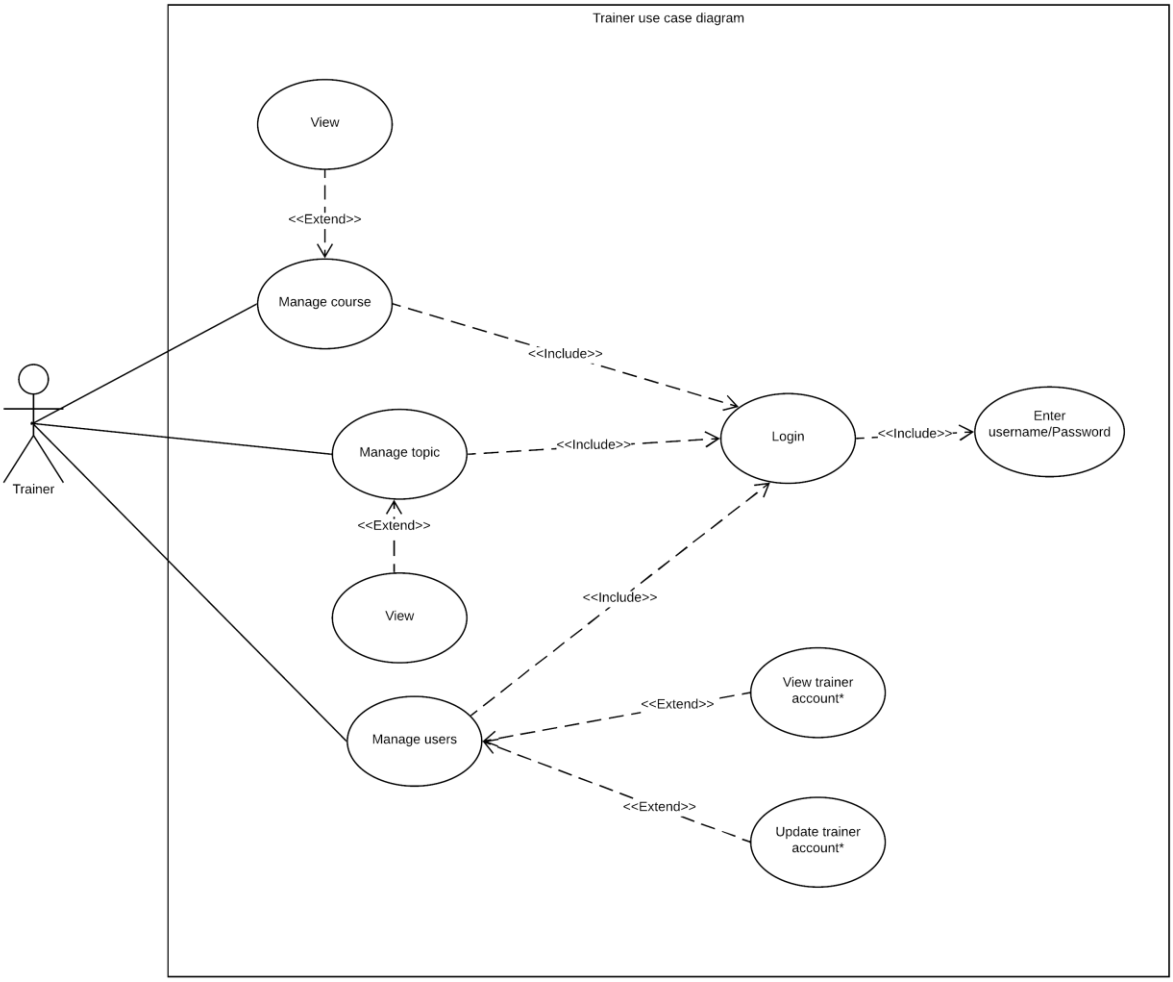


Figure 5. Trainer use case diagram

- **Purpose**: The interaction between Trainer and system
- **Requirements Traceability:** Base on system requirement that system can used to manage trainee accounts, manage trainers, manage courses, manage topics, can login to the system.
- **Priority**: High
- **Actors**: Training Staff
- **Extends:**
    - View/Update* Trainer account
    - View Course
    - View Topic
- **Includes**:
    - Verify username/password

3.3.4 Trainee use case diagram

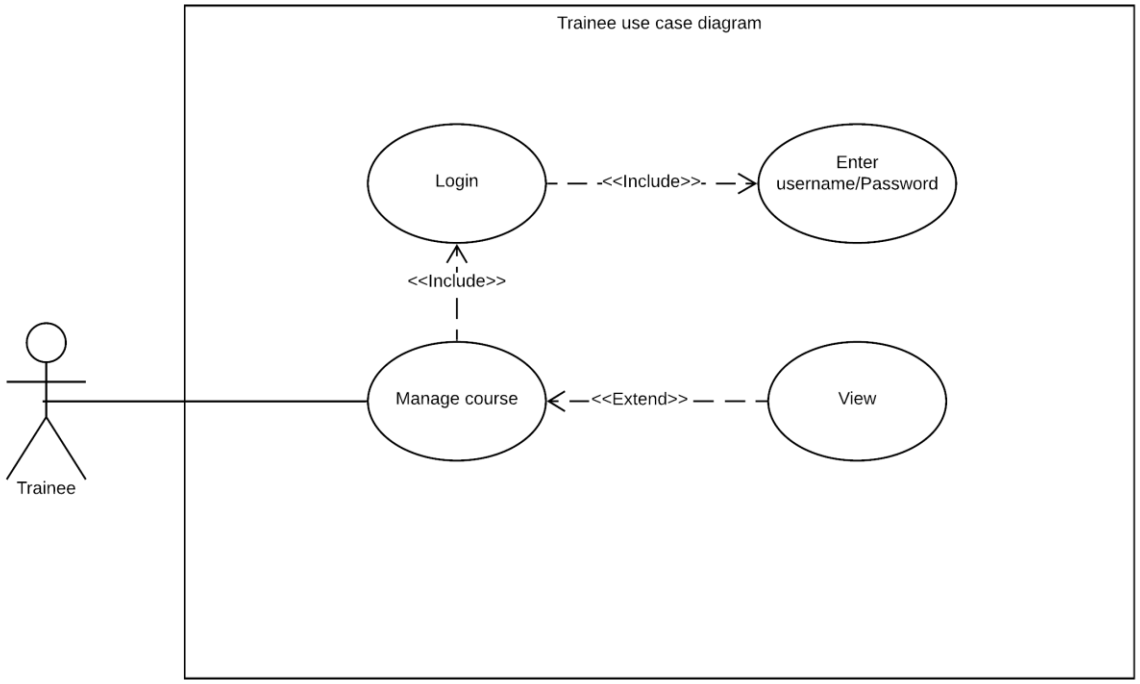


Figure 6. Trainee use case diagram

- **Purpose**: The interaction between Trainee and system.
- **Requirements Traceability:** Base on system requirement that system can used to manage trainee accounts, manage courses, manage topics, can login to the system
- **Priority**: High
- **Actors**: Training Staff
- **Extends:**
    - View Course
- **Includes**:
    - Verify username/password

4 Other Non-functional Requirements

4.1 Performance Requirements

- **First Contentful Paint**: Less or equal to 2.0 s
- **First Meaningful Paint**: Less or equal to 2.0 s
- **Speed Index**: Less or equal to 2.0 s
- **First CPU Idle**: Less or equal to 2.0 s
- **Time to Interactive**: Less or equal to 2.3 s
- **Max Potential First Input Delay**: Less or equal to 60 ms

0.1 second: the limit after which the system reaction doesn't seem instantaneous;
1 second: when user will notice the delay, but without interrupting the flow of thought;
10 seconds: when user attention is completely lost.

The landing page supporting 5 thousand users per hour must provide 6 seconds or less response time in a Chrome desktop browser, including the rendering of text and images, over an LTE connection.

4.2 Safety and Security Requirements

The website will deploy with HTTPS and has SSL

4.3 Software Quality Attributes

The web dashboard must be available to users 98 percent of the time every month during business hours

4.4 Localization

The date format must be as follows: month.date.year.
The font on the website must be easy to read.