FPT HR Learning System

Group 4:
Pham Xuan Hai
Bui Nguyen Ngoc Han
Nguyen Ha Thai Thinh
Ngo Thi Khanh Chi
Than Thi Bich Hang
Truong Hoang Anh Dung

FPT Training

Contents

- INTRODUCTION
- REQUIREMENT SPECIFICATION
- DATABASE
- DEPLOYMENT
- DEVELOPMENT
- CONCLUSION

INTRODUCTION

The management system for a company's internal training program is a software application designed to manage "Training" activities. The Internal Training Management System aims to provide a complete and effective platform to organize, track, and evaluate the company's internal training process.

The system provides a user-friendly interface to manage all training activities from a single platform, improving operational efficiency and minimizing manual errors. It is designed to process large volumes of data, provide real-time insights into training activities, and enable companies to make data-driven decisions.

FPT Training

REQUIREMENT SPECIFICATION

PROJECT SUMARY

FPT Co. aims to establish a comprehensive internal training system to foster continuous learning. The company requires a web-based platform for managing training activities. The system, designated for use by the HR department, encompasses three key roles: Administrator, Training Staff, and Trainer.

FPT Training

Functional Requirements

Login

This function is intended to authenticate users when interacting with the system to provide rights and scope of access to the system.

Trainee Account Management

This function allows administrators to add, edit, and delete information in the system, assign or change trainee usernames and passwords, and search for students.

Trainer Management

This function allows administrators to add, edit, and delete information in the system and assign or change trainer usernames and passwords.

Category Management

Allows administrators to add, edit, delete and search categories in the system.

Course Management

Allows administrators to add, edit, delete and search courses in the system.

Topic management

Allows management of common topics used in courses.

Performance Requirements

Responsiveness: The system should respond to user interactions within 2 seconds under normal load conditions.
Scalability: The system should support a minimum of 100 simultaneous users without performance degradation.
Reliability: The system should have an uptime of 99.9%.
Security: The system should be able to handle authentication and authorization processes.
Real-time Updates: The system should update information in real-time.
Backup and Recovery: System backups should be completed daily, and data recovery should be achievable in case of system failure.

FPT Training

Safety and Security Requirements

Data Privacy Compliance: The system must comply with relevant data privacy regulations such as GDPR or any other applicable laws.

Access Control: Access to training materials and user data must be role-based and restricted to authorized personnel only.

User Authentication: Users must authenticate themselves before accessing the system or any training-related data.

Data Backup and Recovery: Regular automated backups of training data must be performed, with offsite storage for disaster recovery purposes.

Security Compliance Certifications: The system must obtain relevant security certifications (e.g., ISO 27001) to demonstrate compliance with industry standards.

Data Integrity Checks: Implement mechanisms to verify the integrity of training data, detecting and preventing unauthorized modifications or tampering.

Training Content Security: Ensure that training materials are free from malware or malicious content.

Software Quality Attributes

Usability: The system should be intuitive and easy to navigate for users of varying technical backgrounds. Clear interfaces, straightforward workflows, and user-friendly features will enhance overall usability. So need to achieve a minimum score of 85% on user satisfaction surveys conducted, based on responses regarding ease of navigation, clarity of interfaces, and intuitiveness of workflows.

Flexibility: The system should be flexible enough to accommodate various types of training activities, content formats, and scheduling requirements. This flexibility ensures that the system can adapt to the evolving needs of the organization's training program.

Interoperability: Integration capabilities with existing systems or tools used by the company for HR management, employee databases, or learning management systems (LMS) will be important. So need to ensure seamless integration with existing HR systems and learning management systems, with a documented success rate of 99% for data exchange between systems based on performance reports.

Maintainability: The system should be designed with maintainability in mind, including well-structured code, documentation, and modular architecture. This ensures that future updates, enhancements, and bug fixes can be implemented efficiently without disrupting ongoing operations. So need to maintain an average code complexity score below 20 (as measured by cyclomatic complexity) across all modules, and ensure that documentation covers at least 90% of the codebase, verified through regular code reviews and audits.

Reliability: Users rely on the system to access critical training resources and information. Therefore, it's essential that the system operates reliably without frequent downtime or errors. It needs to achieve a system uptime of 99.9% over the course of a year, with no more than 0.1% of user transactions resulting in errors or system failures.

Software Quality Attributes

Testability: Maintain test coverage of at least 85% for all system components, measured through automated testing tools and verified through regular regression testing during each development cycle.
Availability: Ensure that the system is available 24/7, with scheduled downtime limited to no more than 1 hour per month for maintenance purposes, and achieve a mean time to recovery (MTTR) of less than 15 minutes for unplanned outages. This requires reliable hosting infrastructure, disaster recovery plans, and proactive monitoring to minimize downtime and ensure continuous availability.
Adaptability: Implement at least 80% of requested feature enhancements, based on prioritization by stakeholders and tracked through sprint retrospectives and feature completion reports. Regular updates and enhancements will help keep the system aligned with the company's goals and objectives.
Portability: Ensure that the system can be deployed on both on-premises servers and cloud platforms, with documented deployment scripts and configurations available, and achieve a deployment success rate of 95% across different environments, verified through deployment logs and post-deployment checks.
Reusability: Design the components or modules in the system in a reusable manner and ensure that reusable components are used in at least 70% of the new feature implementation to enable enhancements. Increase development efficiency and reduce duplication of effort.

Risk Assessment

Lack of support from end users:

- Risk: End users refuse to accept the new system, making it difficult to deploy and use.
- Solution: Ensure end-user support and involvement from design to implementation. Organize training sessions and facilitate early feedback from their side.

Lack of resources or skills:

- Risk: Lack of resources or skills needed to develop and deploy the system.
- Solution: Assess and identify the skills and resources needed from the early stages of the project. Consider hiring or training staff, or using an outside service if needed.

Lack of uniformity in project requirements:

- Risk: Project requirements are not clearly defined or change constantly, leading to ambiguity during development.
- Solution: Establish and maintain a process to identify and manage project requirements. Regularly interact with customers and users to identify and agree on requirements.

Technical problems or software errors:

- Risk: Technical problems or software errors cause interruptions during deployment and use.
- Solution: Perform thorough technical testing and error checking before deployment. Establish an effective error management process to track, report, and fix errors that arise.

Information security issues:

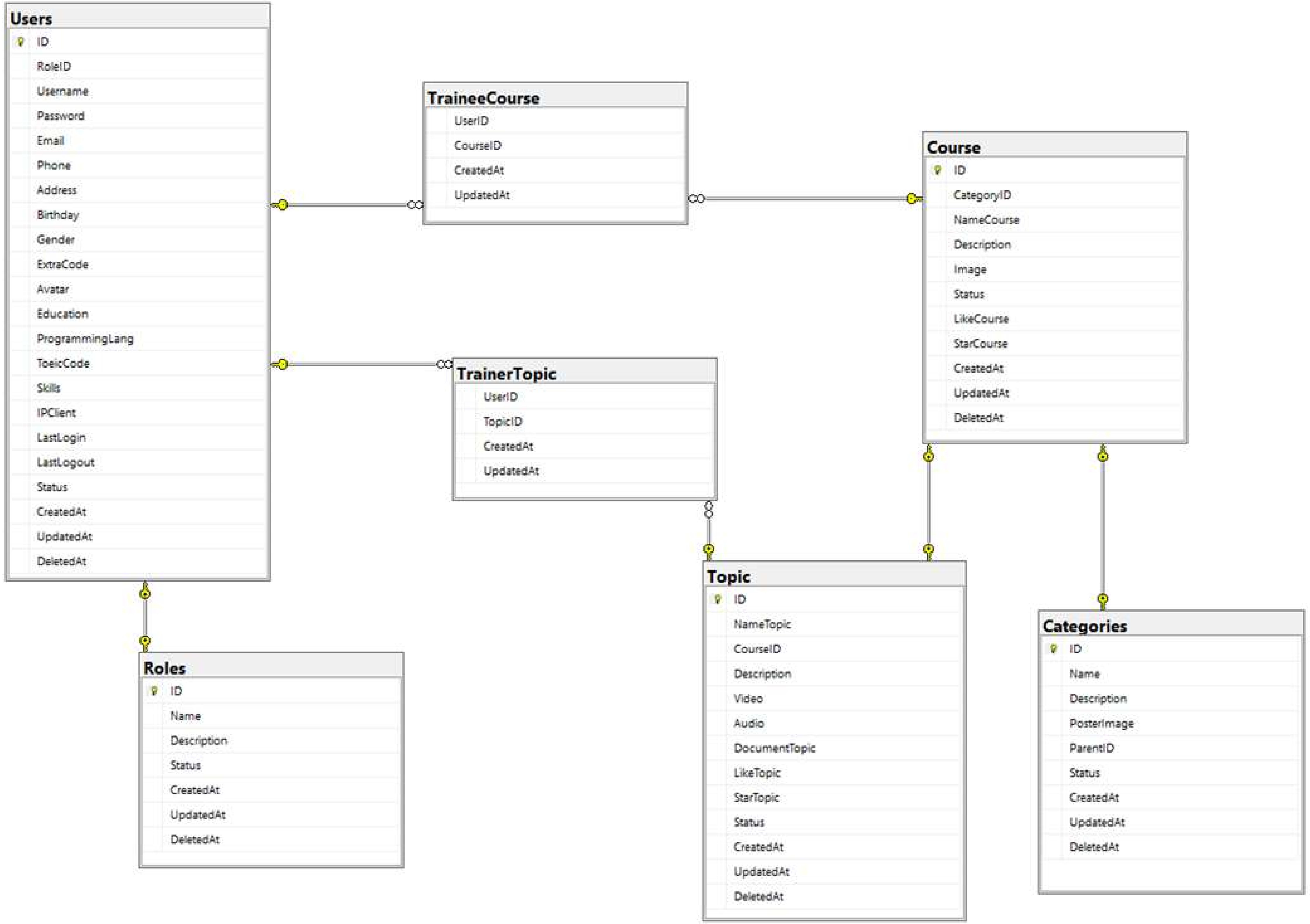
- Risk: Important or sensitive data could be stolen or leaked.
- Solution: Implement information security measures such as data encryption, periodic checks and software updates, as well as train employees on information security rules.

FPT Training

DATABASE

FPT Training

Database diagram



Users
ID
RoleID
Username
Password
Email
Phone
Address
Birthday
Gender
ExtraCode
Avatar
Education
ProgrammingLang
ToeicCode
Skills
IPClient
LastLogin
LastLogout
Status
CreatedAt
UpdatedAt
DeletedAt

TraineeCourse
UserID
CourseID
CreatedAt
UpdatedAt

Course
ID
CategoryID
NameCourse
Description
Image
Status
LikeCourse
StarCourse
CreatedAt
UpdatedAt
DeletedAt

TrainerTopic
UserID
TopicID
CreatedAt
UpdatedAt

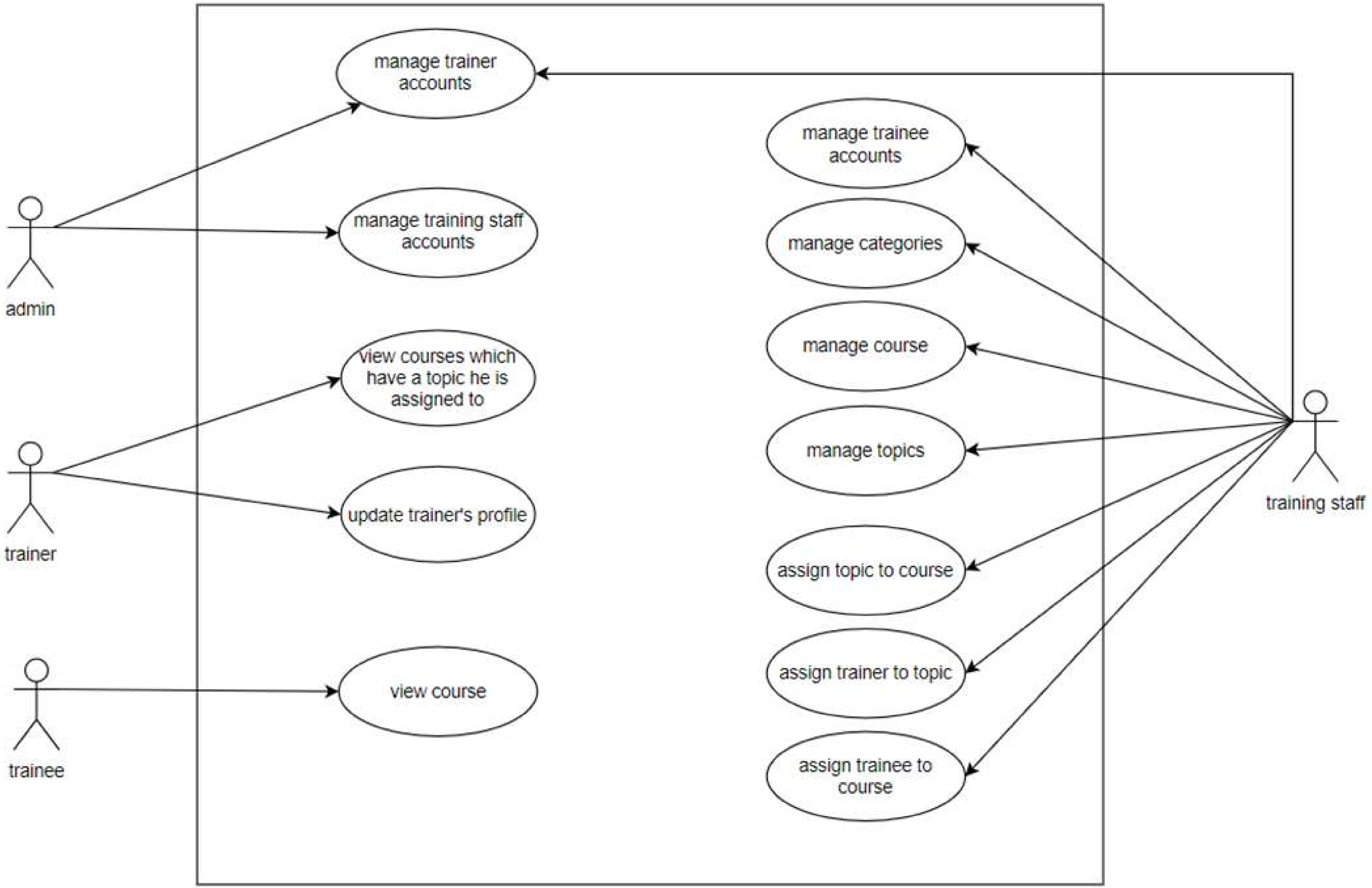
Roles
ID
Name
Description
Status
CreatedAt
UpdatedAt
DeletedAt

Topic
ID
NameTopic
CourseID
Description
Video
Audio
DocumentTopic
LikeTopic
StarTopic
Status
CreatedAt
UpdatedAt
DeletedAt

Categories
ID
Name
Description
PosterImage
ParentID
Status
CreatedAt
UpdatedAt
DeletedAt

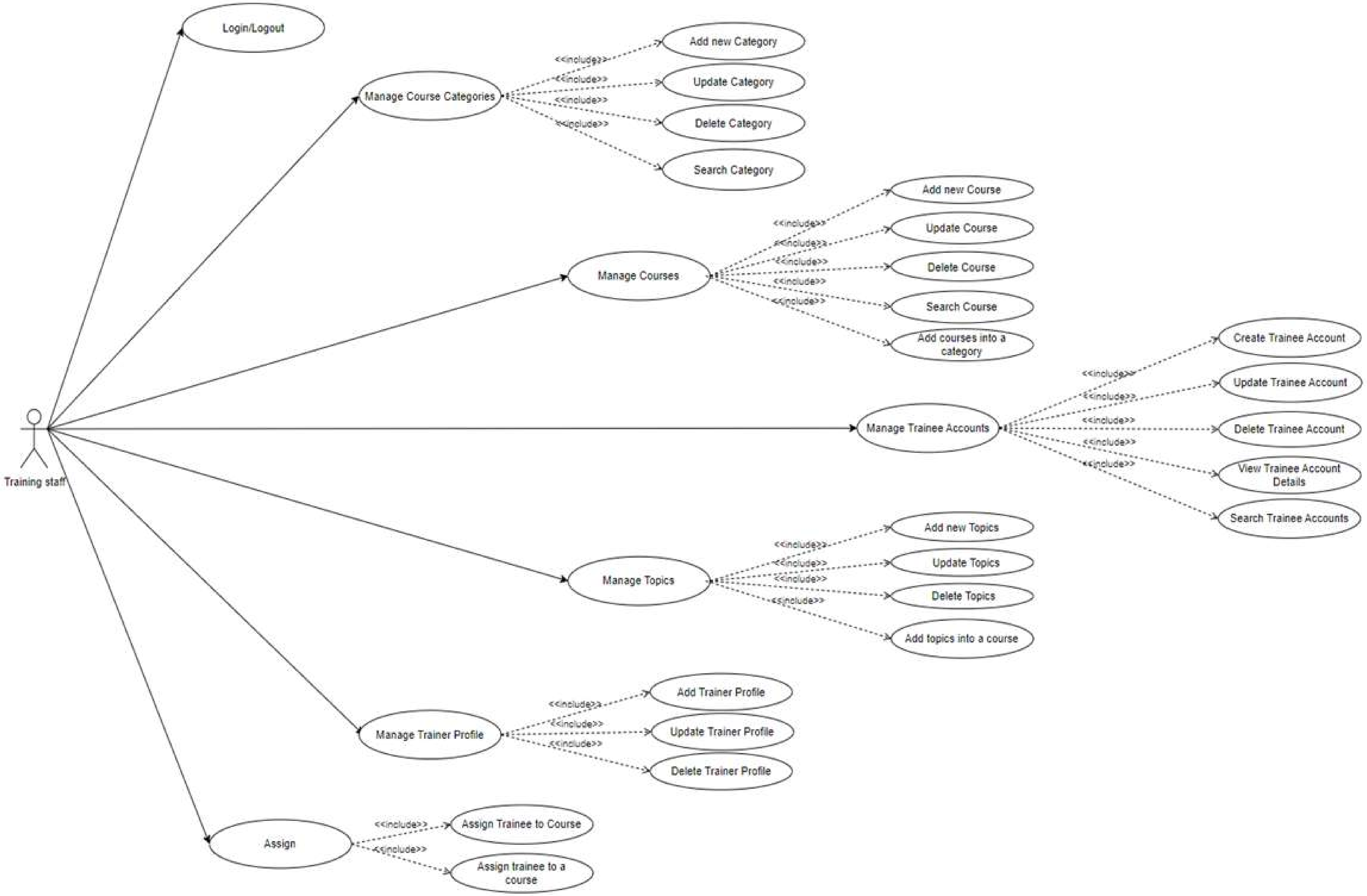
FPT Training

Usecase diagram



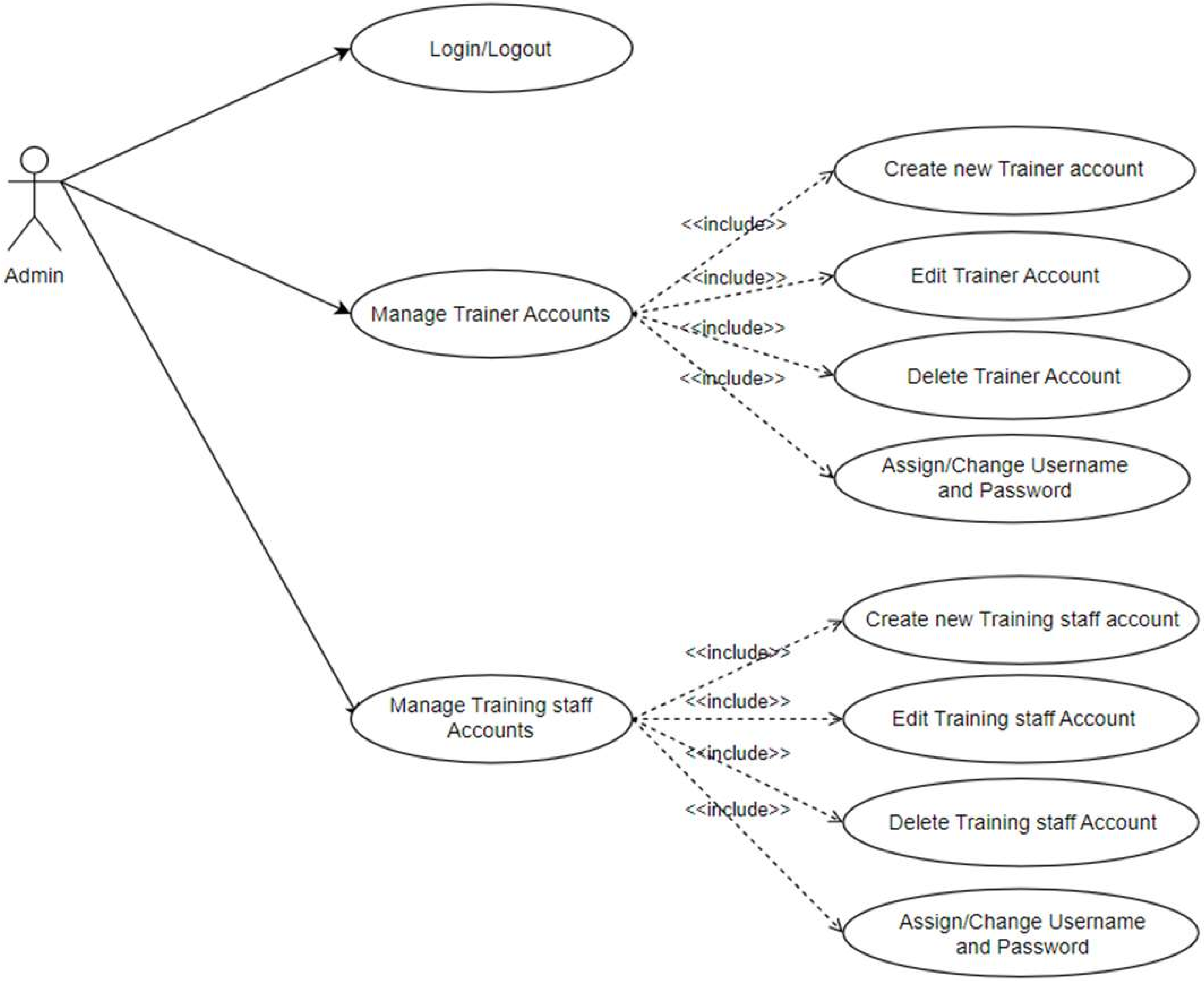
FPT Training

Training staff usecase diagram



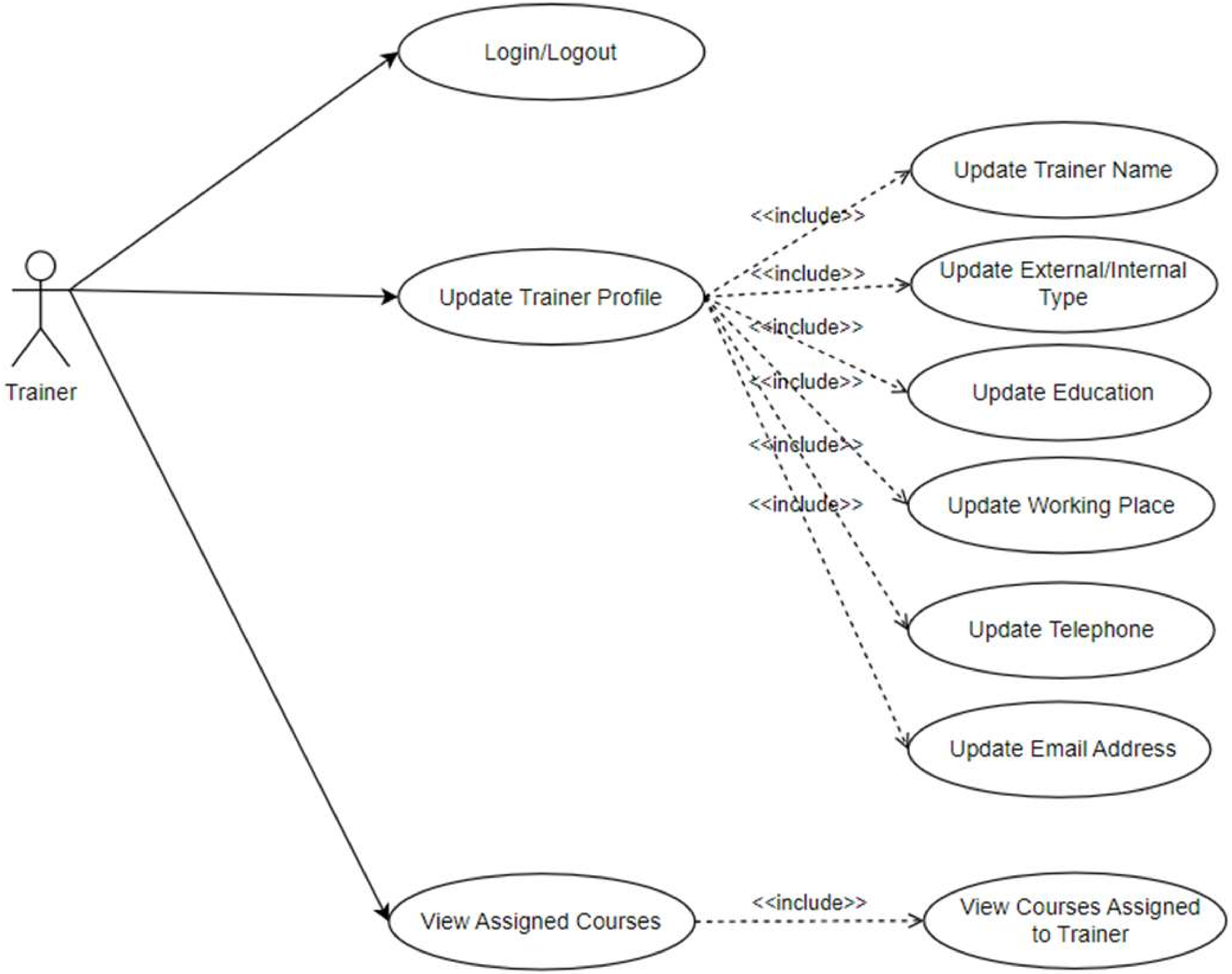
FPT Training

Admin usecase diagram

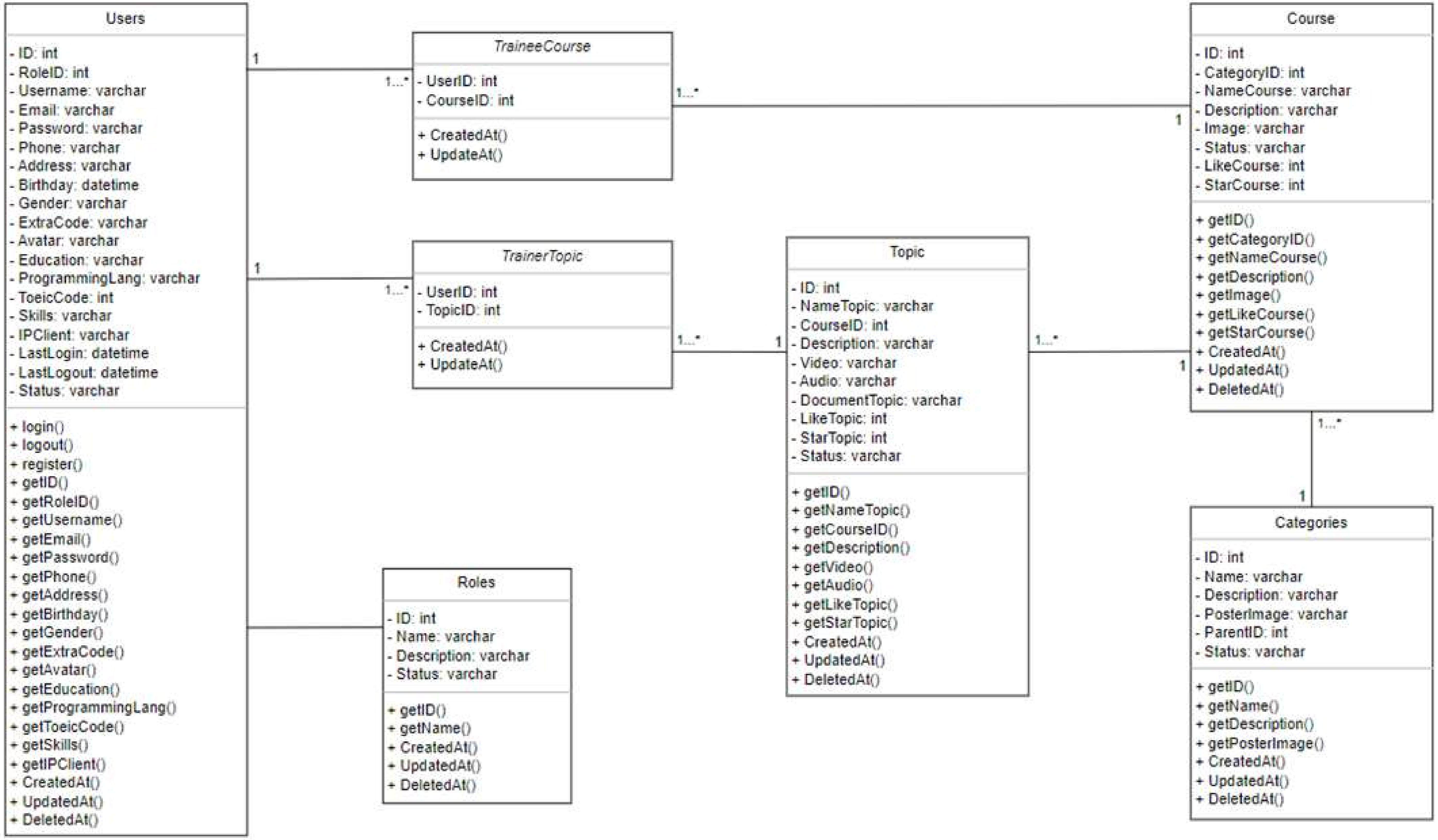

Admin
Login/Logout
Manage Trainer Accounts
<<include>>
<<include>>
<<include>>
<<include>>
Create new Trainer account
Edit Trainer Account
Delete Trainer Account
Assign/Change Username and Password
Manage Training staff Accounts
<<include>>
<<include>>
<<include>>
<<include>>
Create new Training staff account
Edit Training staff Account
Delete Training staff Account
Assign/Change Username and Password

FPT Training

Trainer usecase diagram



UML



Users

- ID: int
- RoleID: int
- Username: varchar
- Email: varchar
- Password: varchar
- Phone: varchar
- Address: varchar
- Birthday: datetime
- Gender: varchar
- ExtraCode: varchar
- Avatar: varchar
- Education: varchar
- ProgrammingLang: varchar
- ToeicCode: int
- Skills: varchar
- IPClient: varchar
- LastLogin: datetime
- LastLogout: datetime
- Status: varchar

+ login()
+ logout()
+ register()
+ getID()
+ getRoleID()
+ getUsername()
+ getEmail()
+ getPassword()
+ getPhone()
+ getAddress()
+ getBirthday()
+ getGender()
+ getExtraCode()
+ getAvatar()
+ getEducation()
+ getProgrammingLang()
+ getToeicCode()
+ getSkills()
+ getIPClient()
+ CreatedAt()
+ UpdatedAt()
+ DeletedAt()

TraineeCourse

- UserID: int
- CourseID: int

+ CreatedAt()
+ UpdateAt()

TrainerTopic

- UserID: int
- TopicID: int

+ CreatedAt()
+ UpdateAt()

Roles

- ID: int
- Name: varchar
- Description: varchar
- Status: varchar

+ getID()
+ getName()
+ CreatedAt()
+ UpdatedAt()
+ DeletedAt()

Topic

- ID: int
- NameTopic: varchar
- CourseID: int
- Description: varchar
- Video: varchar
- Audio: varchar
- DocumentTopic: varchar
- LikeTopic: int
- StarTopic: int
- Status: varchar

+ getID()
+ getNameTopic()
+ getCourseID()
+ getDescription()
+ getVideo()
+ getAudio()
+ getLikeTopic()
+ getStarTopic()
+ CreatedAt()
+ UpdatedAt()
+ DeletedAt()

Course

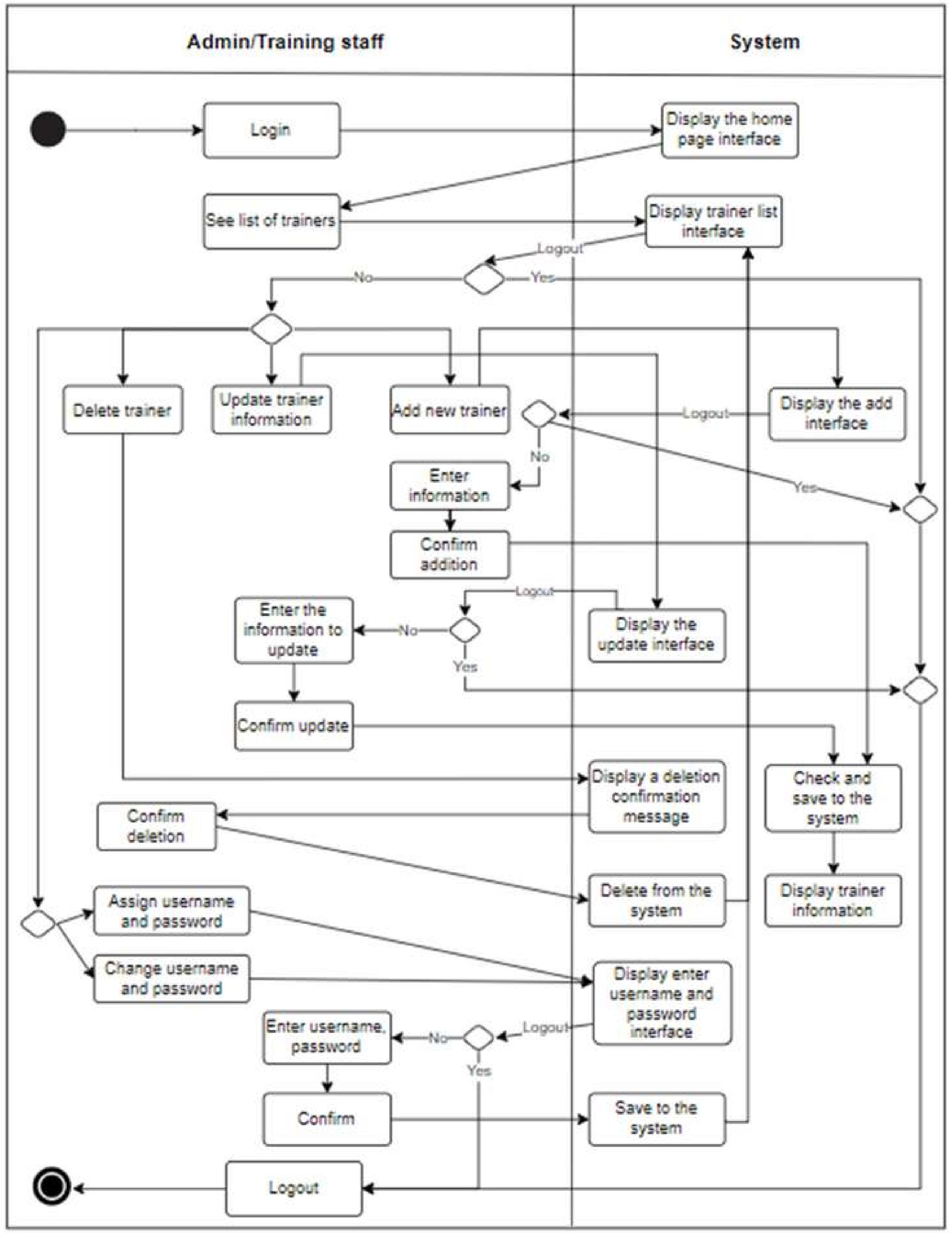
- ID: int
- CategoryID: int
- NameCourse: varchar
- Description: varchar
- Image: varchar
- Status: varchar
- LikeCourse: int
- StarCourse: int

+ getID()
+ getCategoryID()
+ getNameCourse()
+ getDescription()
+ getImage()
+ getLikeCourse()
+ getStarCourse()
+ CreatedAt()
+ UpdatedAt()
+ DeletedAt()

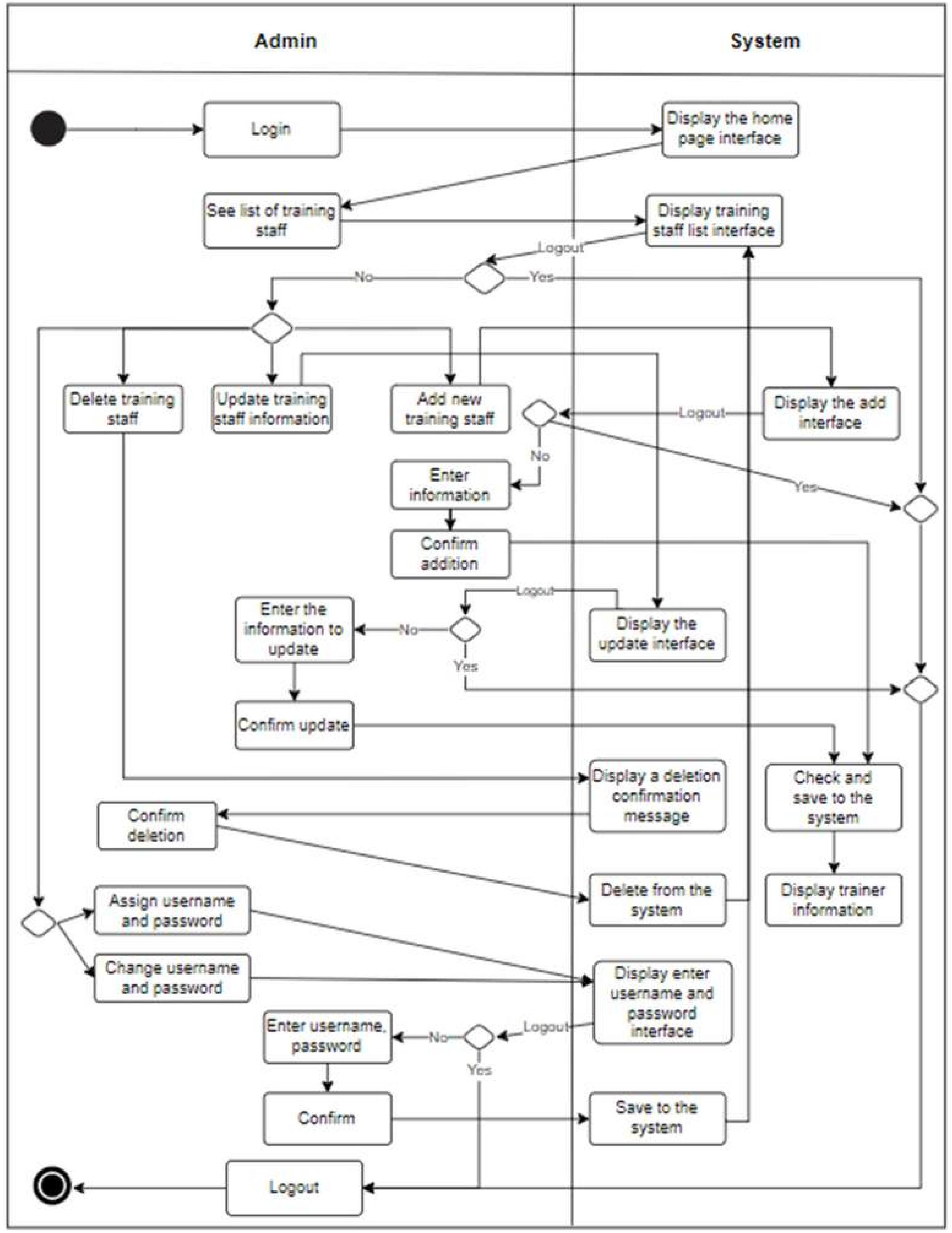
Categories

- ID: int
- Name: varchar
- Description: varchar
- PosterImage: varchar
- ParentID: int
- Status: varchar

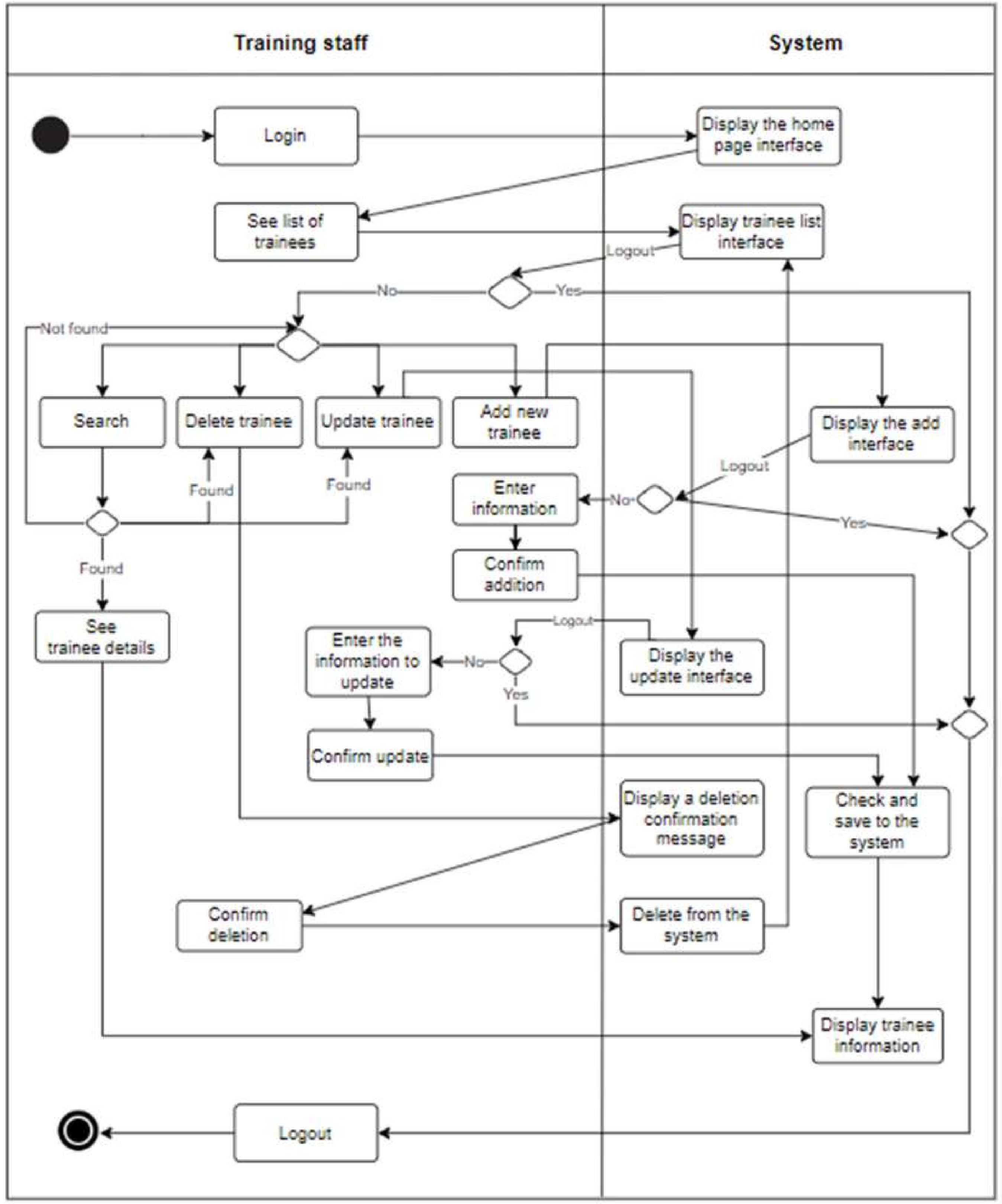
+ getID()
+ getName()
+ getDescription()
+ getPosterImage()
+ CreatedAt()
+ UpdatedAt()
+ DeletedAt()


Admin/Training staff
System
Login
Display the home page interface
See list of trainers
Display trainer list interface
Logout
No
Yes
Delete trainer
Update trainer information
Add new trainer
Display the add interface
Logout
No
Yes
Enter information
Confirm addition
Logout
Enter the information to update
No
Yes
Display the update interface
Confirm update
Display a deletion confirmation message
Check and save to the system
Confirm deletion
Delete from the system
Display trainer information
Assign username and password
Change username and password
Display enter username and password interface
Enter username, password
No
Logout
Yes
Confirm
Save to the system
Logout

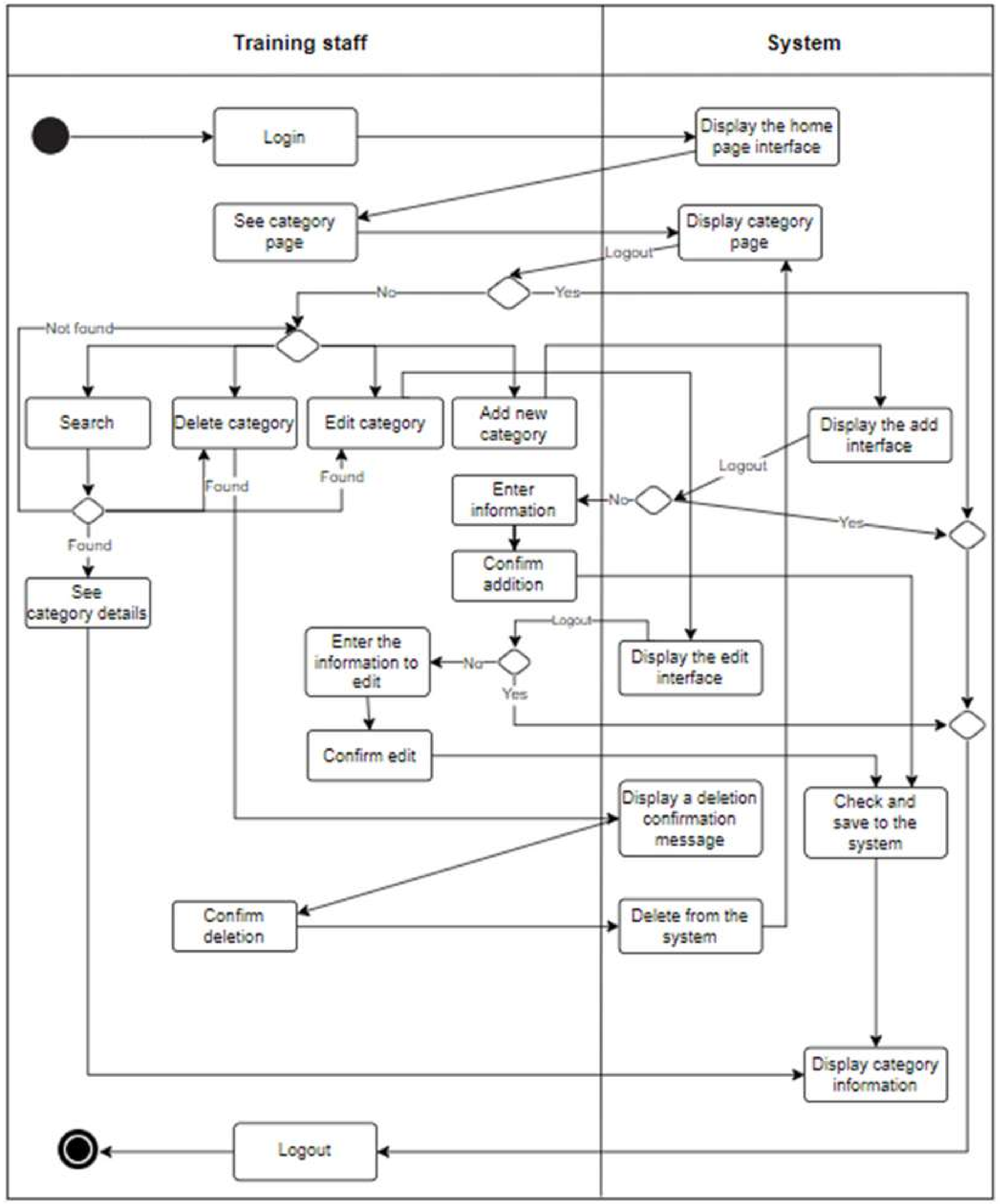
Trainer account management process


Admin
System
Login
Display the home page interface
See list of training staff
Display training staff list interface
Logout
No
Yes
Delete training staff
Update training staff information
Add new training staff
Display the add interface
Logout
No
Yes
Enter information
Confirm addition
Logout
Enter the information to update
No
Yes
Display the update interface
Confirm update
Display a deletion confirmation message
Check and save to the system
Confirm deletion
Delete from the system
Display trainer information
Assign username and password
Change username and password
Display enter username and password interface
Enter username, password
No
Logout
Yes
Confirm
Save to the system
Logout

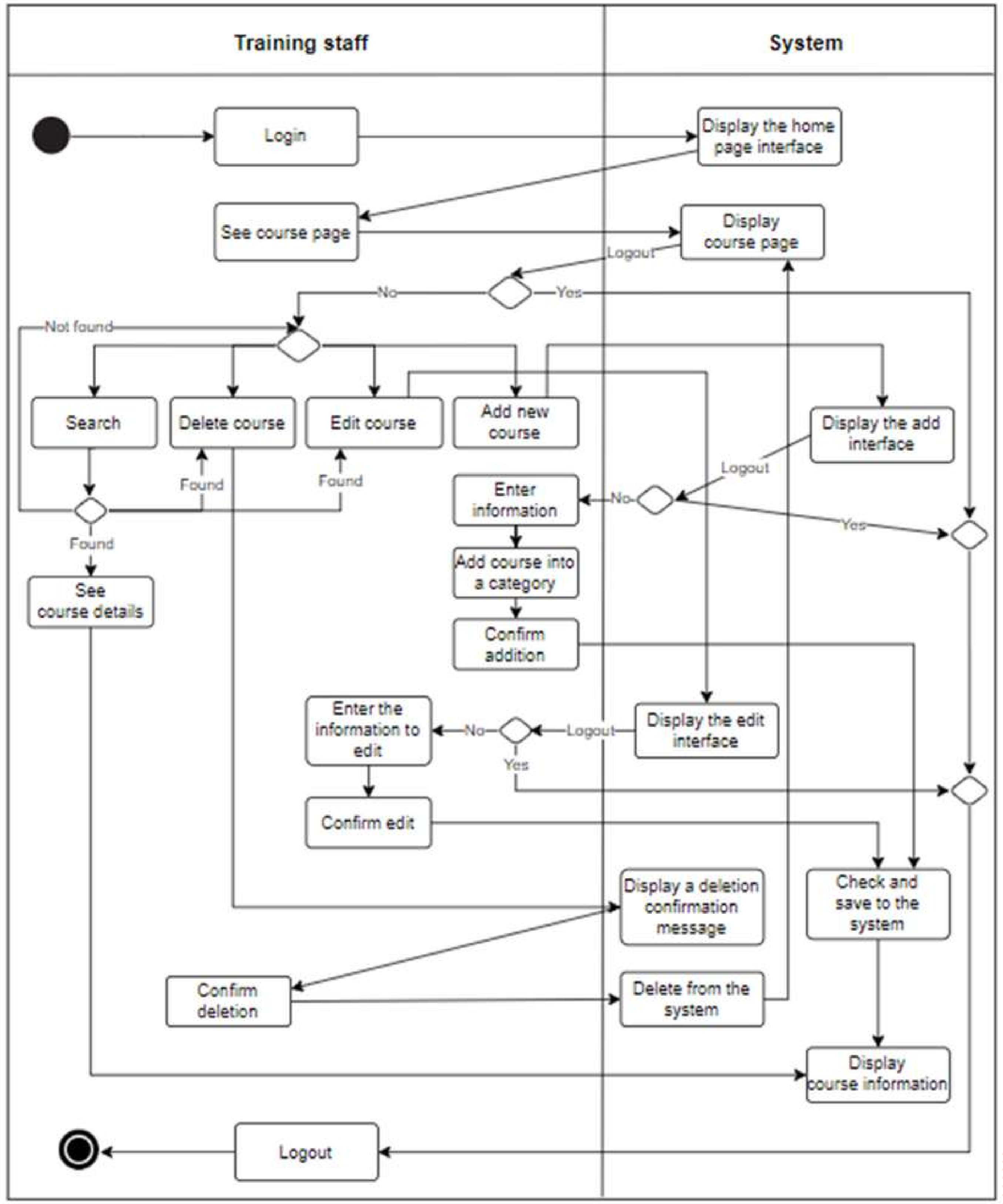
Training staff account management process



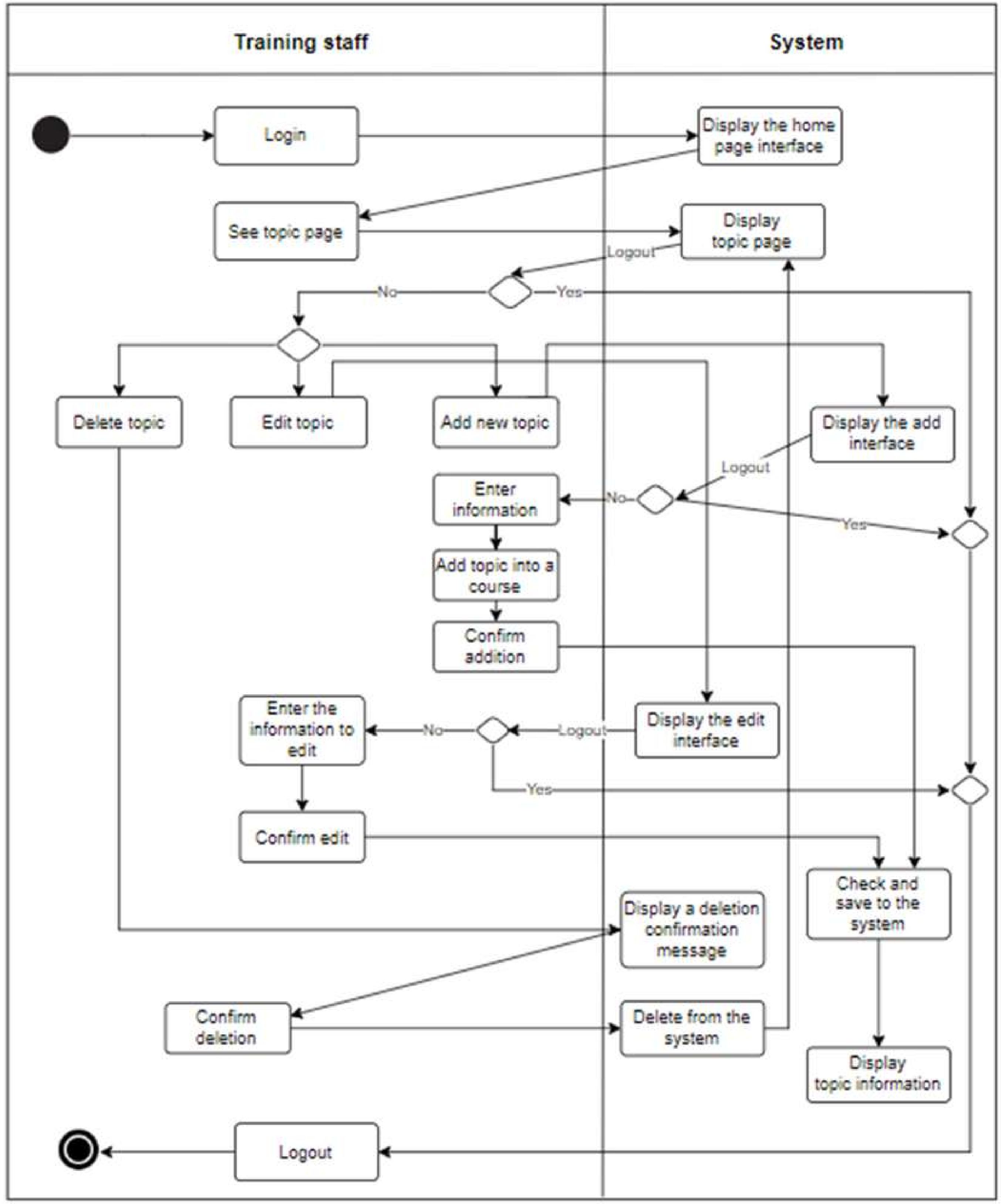
Trainee account management process



Category management process



Course management process



Topic management process

Assumptions and Dependencies

Assumptions:

Employee Resistance to Change: There might be resistance among employees to adopt a new training management system, especially if they are accustomed to existing processes or tools. This resistance could influence the requirements for user interface design, training and support resources, and change management strategies.

Budget Constraints: Assumptions about the budget allocated for the project could impact the scope and features of the training management system. Limited resources may necessitate prioritization of certain functionalities over others.

Technology Compatibility: Assumptions regarding the existing technology infrastructure within the organization could affect the technical requirements for the training management system. Compatibility with legacy systems or integration with other software tools may need to be considered.

Organizational Culture: The organizational culture, including attitudes towards technology adoption, collaboration, and communication, could influence the requirements for the system. For example, if the company values decentralized decision-making, the system may need to accommodate multiple stakeholders with varying levels of authority.

Assumptions and Dependencies

Assumptions:

Data Privacy and Security Concerns: Assumptions about data privacy regulations and security requirements could impact the design and implementation of the system. Compliance with industry standards or legal frameworks may influence features such as user authentication, data encryption, and access controls.

Scalability and Growth: Assumptions about the future growth and scalability of the organization could influence the requirements for the system. The system may need to be scalable to accommodate an increasing number of users, training activities, and organizational changes over time.

Stakeholder Expectations: Assumptions about the expectations and preferences of key stakeholders, such as HR managers, trainers, and employees, could impact the requirements for the system. Gathering input from stakeholders and managing their expectations throughout the project lifecycle is crucial for success.

External Events or Disruptions: Assumptions about external events or disruptions, such as changes in market conditions, regulatory requirements, or technological advancements, could impact the requirements for the system. The system may need to be adaptable and resilient.

Assumptions and Dependencies

Dependencies:

Third-Party Libraries or Frameworks: If the project plans to utilize third-party libraries or frameworks for specific functionalities such as user authentication, data visualization, or reporting, there might be dependencies on the availability, compatibility, and licensing terms of these components.

APIs and Integration Points: The training management system may need to integrate with existing systems or external services for functionalities such as management or reporting. Dependencies on the stability and accessibility of these APIs or integration points need to be considered.

Data Sources and Data Feeds: The system may rely on data from external sources or feeds, such as employee records, training content repositories, or performance metrics. Dependencies on the availability, accuracy, and consistency of these data sources need to be addressed.

Infrastructure and Hosting Services: Dependencies on external infrastructure or hosting services, such as cloud platforms or data centers, may exist for hosting the application, databases, or oter components of the system. Considerations include reliability, scalability, and performance of these services.

Regulatory Compliance and Standards: The project may have dependencies on compliance with industry standards, regulations, or legal requirements, such as data privacy regulations (e.g., GDPR) or accessibility standards (e.g., WCAG). Adherence to these standards may impact the design and implementation of the system.

Internal Policies and Procedures: Dependencies on internal policies, procedures, and guidelines within the organization, such as IT security policies, change management processes, or procurement policies, need to be considered when developing the system.

FPT Training

DEPLOYMENT

Step 1: Prepare the environment:

- Set up the deployment environment, ensuring it meets system requirements. In this project we choose Visual Studio 2022 to deploy the project.

Step 2: Deploy the database:

- Back up and deploy the system's database to the deployment environment. We choose SQLServer to deploy this project.

Step 3: Deploy the application:

- Back up and deploy application code to the deployment environment.

Step 4: Configuration and integration:

- Configure environment variables and integrate the system with the HR database. Step 5: Testing in phase:
- Conduct testing in a staging environment to ensure a smooth transition.

Step 6: Safe deployment:

- Implement security measures, such as HTTPS, for a secure deployment.

Step 7: User acceptance testing (UAT):

- Conduct user acceptance testing with HR staff to validate functionality.

DEVELOPMENT

Phase 1: Requirement Analysis

- Conduct in-depth discussions with HR department to understand specific needs.
- Define system requirements, functionalities, and constraints.
- Document user roles (Administrator, Training Staff, Trainer).
- Develop a detailed Requirement Specification Document.

Phase 2: Planning and Design

- Plan the development timeline, resource allocation, and budget.
- Create a comprehensive project plan outlining tasks and milestones.
- Design the system architecture, including database structure and user interface.
- Define the data flow and relationships between different components.

FPT Training

DEVELOPMENT

Phase 3: Prototype Development Phase:

- Build a prototype of the system based on the design.
- Validate the prototype with key stakeholders to gather feedback.
- Make necessary adjustments to the system's design and functionalities.

Phase 4: Core System Development Phase:

- Initiate the actual coding and development of the Training Management System.
- Implement features for managing trainee accounts, trainers, course categories, courses, and topics.
- Develop functionalities for assigning topics to courses, trainers to topics, and trainees to courses.

DEVELOPMENT

Phase 5: Testing and Quality Assurance Phase:

- Conduct thorough testing of the developed system.
- Perform unit testing, integration testing, and system testing.
- Identify and rectify any bugs or issues.
- Ensure the system meets the specified requirements and is user-friendly.

Phase 6: Deployment and Integration Phase:

- Prepare the system for deployment to a testing environment.
- Integrate the system with existing HR tools and databases.
- Test the system's performance and functionality in a real-world scenario.
- Address any issues that arise during the deployment and integration process.

FPT Training

DEVELOPMENT

Phase 7: User Training and Acceptance Testing Phase:

- Provide training sessions for administrators, training staff, and trainers.
- Conduct acceptance testing with end-users to ensure the system aligns with their expectations.
- Gather feedback from users and make final adjustments as needed.

Phase 8: Final Deployment and Maintenance Phase:

- Deploy the finalized system to the production environment.
- Implement continuous monitoring and address any post-deployment issues.
- Establish a maintenance plan for regular updates and improvements.ring the deployment and integration process.

FPT Training

CONCLUSION

In short, our SRS provides all the necessary information, images and diagrams for the project. The internal training management system project will provide a complete and effective platform to organize, monitor and evaluate the company's internal training process. This system will include the following functions as requested: Login, Management: Trainer account, Trainee account, Training staff account, category, course, topic, Assign: Topics to Courses, Trainers to Topic, Trainees to Courses.