# ASSIGNMENT 1 FRONT SHEET

| | | | |
|---|---|---|---|
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | Unit 30: Application Development | | |
| **Submission date** | | **Date Received 1st submission** | |
| **Re-submission Date** | | **Date Received 2nd submission** | |
| **Student Name** | Nguyen Lam Thang | **Student ID** | GCH18545 |
| **Class** | GCH0715 | **Assessor name** | Truong Cong Doan |
| **Student declaration**<br><br>I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
| | | **Student's signature** | |

**Grading grid**

| P1 | P2 | P3 | M1 | M2 | D1 |
|---|---|---|---|---|---|
| | | | | | |

| ❒ Summative Feedback: | ❒ Resubmission Feedback: | |
|---|---|---|
| | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Lecturer Signature:** | | |

**Group Name: <SSTTH>**


| | | |
|---|---|---|
| **<Nguyen Lam Thang>** | **<GCH18545>** | **<thangnlgch18545@fpt.edu.vn>** |
| **<Nguyen Van Tuan>** | **<GCH18078 >** | **<tuannvgch18078@fpt.edu.vn >** |
| **<Nguyen Minh Son>** | **<GCH18055 >** | **<sonnmgch18055@fpt.edu.vn >** |
| **<Duong Duc Son>** | **<GCH18524>** | **< sonddgch18524@fpt.edu.vn >** |
| **<Le Anh Hieu>** | **<GCH18106 >** | **< hieulagch18106@fpt.edu.vn >** |


| | |
|---|---|
| **Instructor:** | **< Truong Cong Doan >** |
| **Course:** | **<Application Development >** |
| **Lab Section:** | **<place your lab section here>** |
| **Teaching Assistant:** | **<Truong Cong Doan>** |

# Table of Contents

# I. Introduction

In this project, my group develops a system, which manages the activity of "Training" for the internal training program of the company. To make clear this system, my group provides a Software Requirements Specification, include Document Purpose, Product Scope, Definitions, Acronyms and Abbreviations Overall Description, Other Non-functional Requirements, etc.

## 1. Document Purpose

The goal of the project is to build a continuing study environment throughout the corporation. So we decide to create a system which manages the activity of "Training" for internal training program of the company.

The purpose of this document is making clear this system. This system can be 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.

Besides, this is a system used by HR department. There are three roles in this system, an administrator, training staff and a trainer.

## 2. Product Scope

This system is a large system to manage the activity of "Training" for the internal training program of the company. Therefore, the scope of system is internal in the company.

## 3. Intended Audience and Document Overview

The document is intended for IT department, project managers, administrator, training staff and a trainer, testers, and documentation writers.

This SRS contains introduction, over all description, specific requirements, other nonfunctional requirements and two appendixes

| Department | Sections |
|---|---|
| IT | I, II, III, IV |
| Test | I, II, III, IV |
| Users | I, II, III |

### 4. Definitions, Acronyms and Abbreviations

| | Acronyms | Full written words |
|---|---|---|
| | SRS | Software Requirements Specification |

Definitions:

**A software requirements** specification is a description of a software system to be developed. It is modeled after business requirements specification, also known as a stakeholder requirements specification.

### 5. Document Conventions

In general this document follows the following formatting requirements. Use TimeNewRoman font size 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1" margins found in this document.

### 6. References and Acknowledgments

## II. Overall Description

### 1. Product Overview

As the technology is being developed rapidly nowadays, FPT Co. desires to build the continuing study environment throughout the corporation. So a system is created by my group. It manages the activity of "Training" for the internal training program of the company

I will make clear the system interacts through illustrating images:

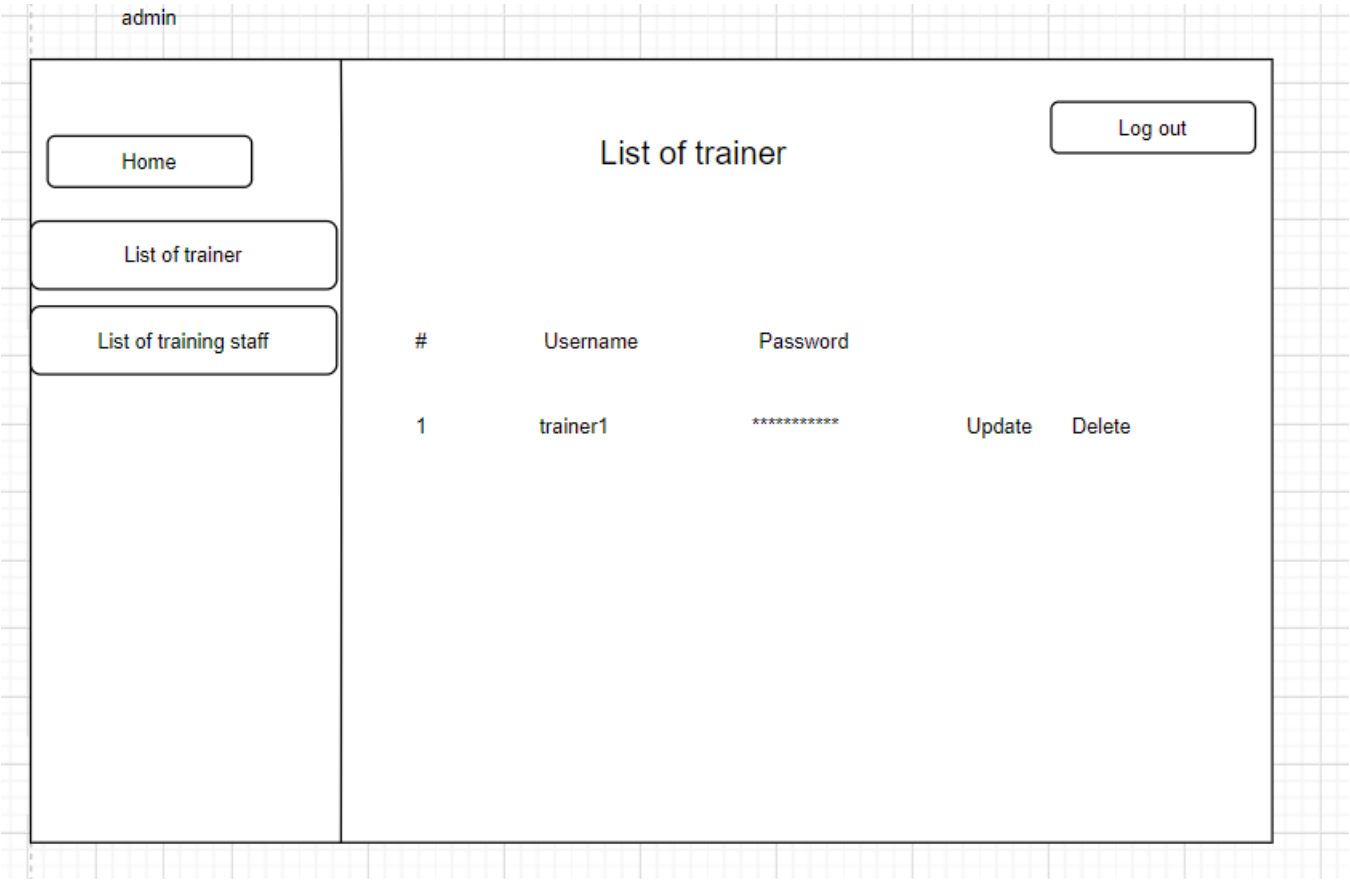



*Figure 1: Admin*

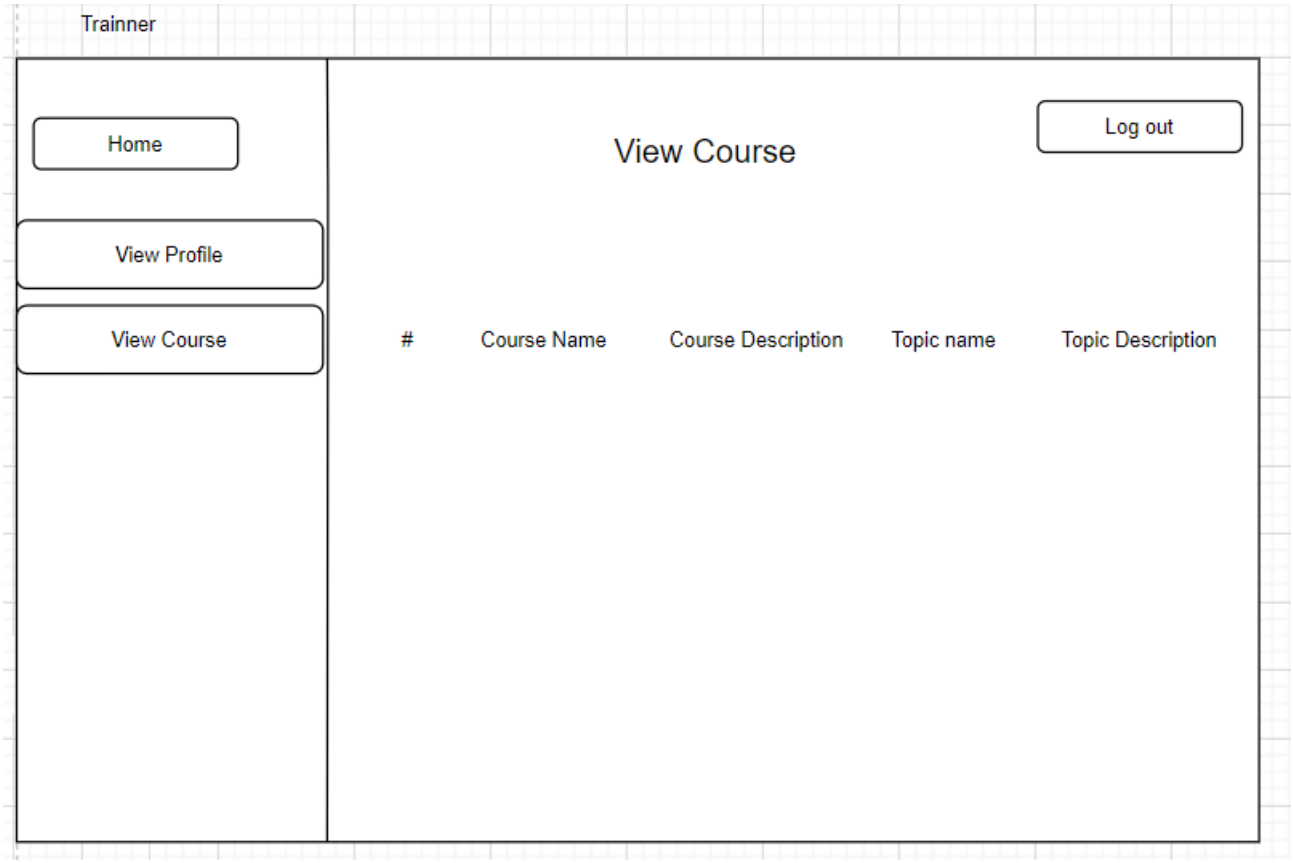



*Figure 2: Trainer*

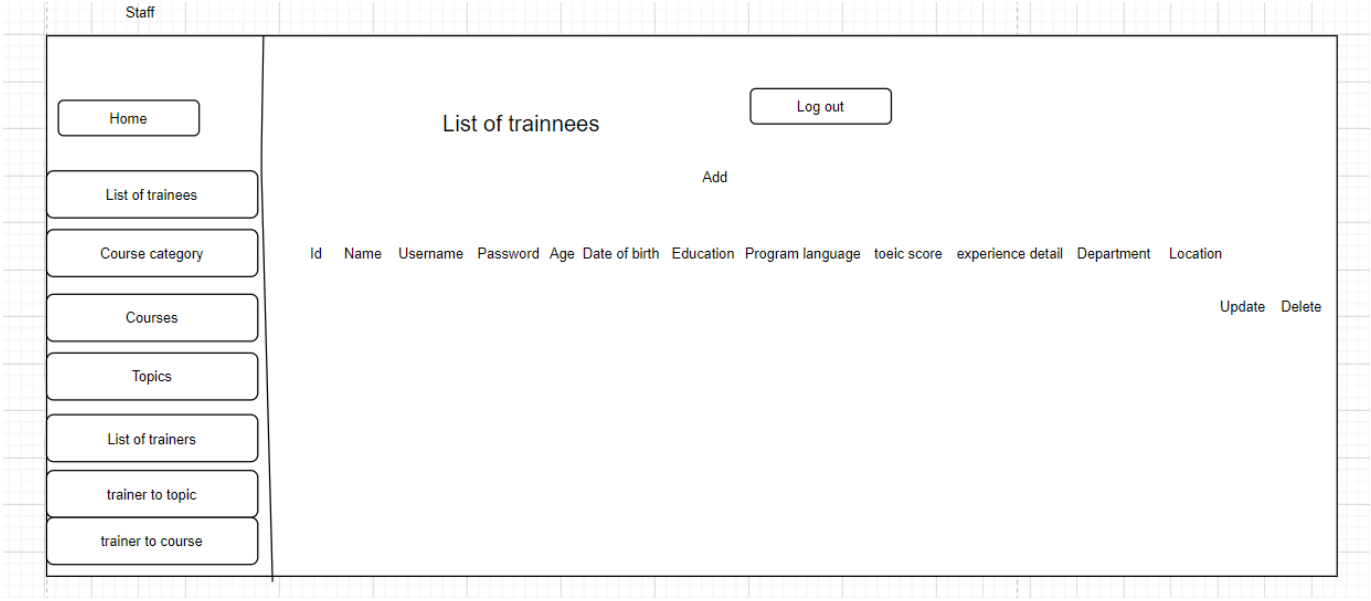



*Figure 3:Training Staff*

## 2. Product Functionality

This is a system used by HR department. Therefore, there are three roles in this system, an administrator, training staff and a trainer.

| Role | Activities |
|---|---|
| Administrator | - Add, delete, edit accounts of Training Staff<br>- Add, delete, edit accounts of Trainer<br>- Login to the system |
| Training staff | - View, search, update, delete accounts of Trainee<br>- Add, delete, search, update, search Course Category<br>- Add, delete, search, update, search Course<br>- Add, delete, search, update, search Course Topic<br>- Login to the system |
| Trainer | - View Courses<br>- Login to the system |

## 3. Design and Implementation Constraints

In the project, there aren't any items or issues that will limit the options available to the developers.

Besides, during developing the system, we decide to use main language that includes COMET method . It's NodeJS and its framework ( Express )

Also, to develop the system, my group used the UML modeling that includes the use case diagram and ERD diagram

## 4. Assumptions and Dependencies

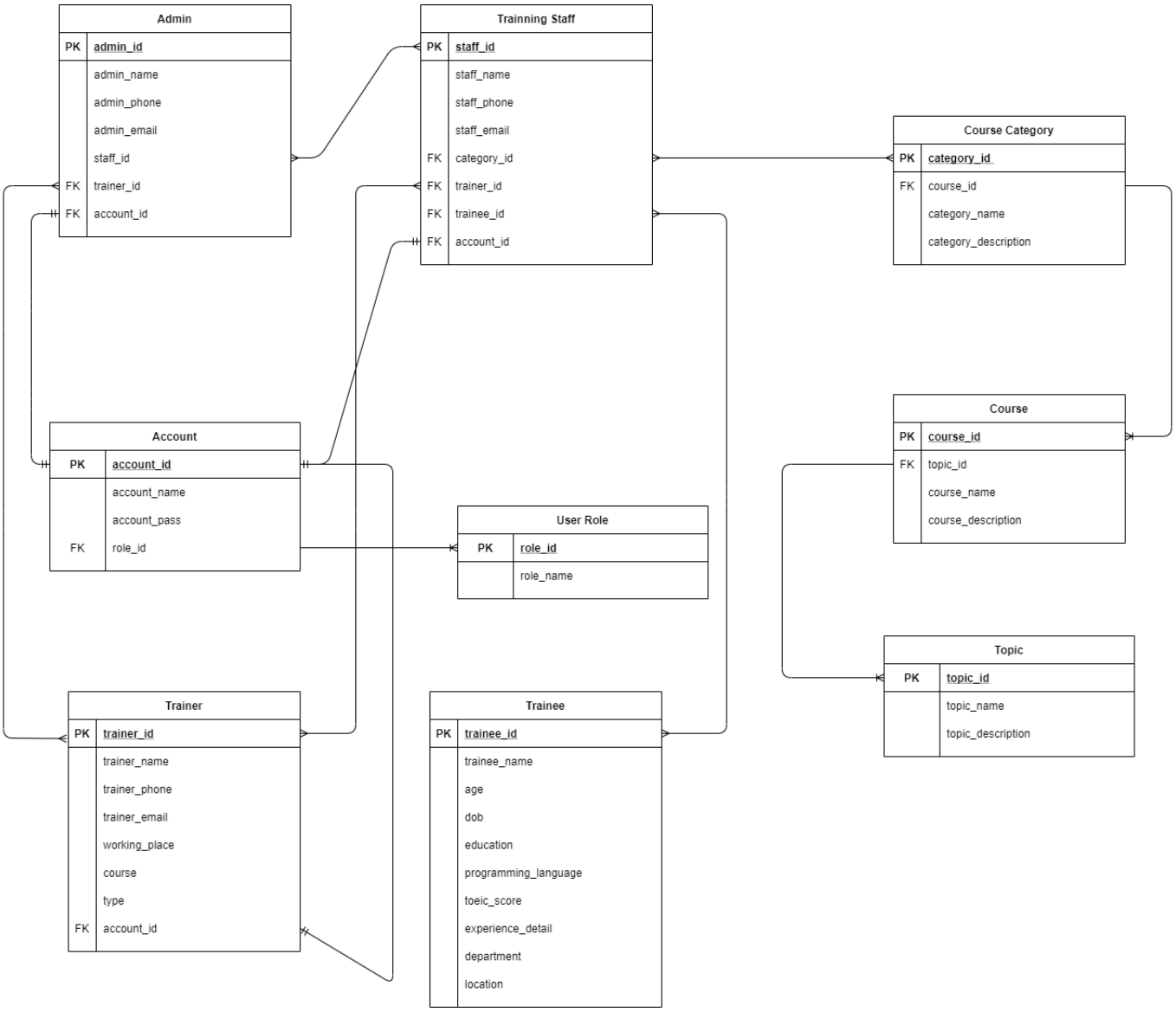
Maybe there are assumed factors that could affect the requirements.

For example, the first factor is the database. Because we use Cloud Computing, the database of the system is hired online. So if there are any problems of third-party, the system definitely has problems too.

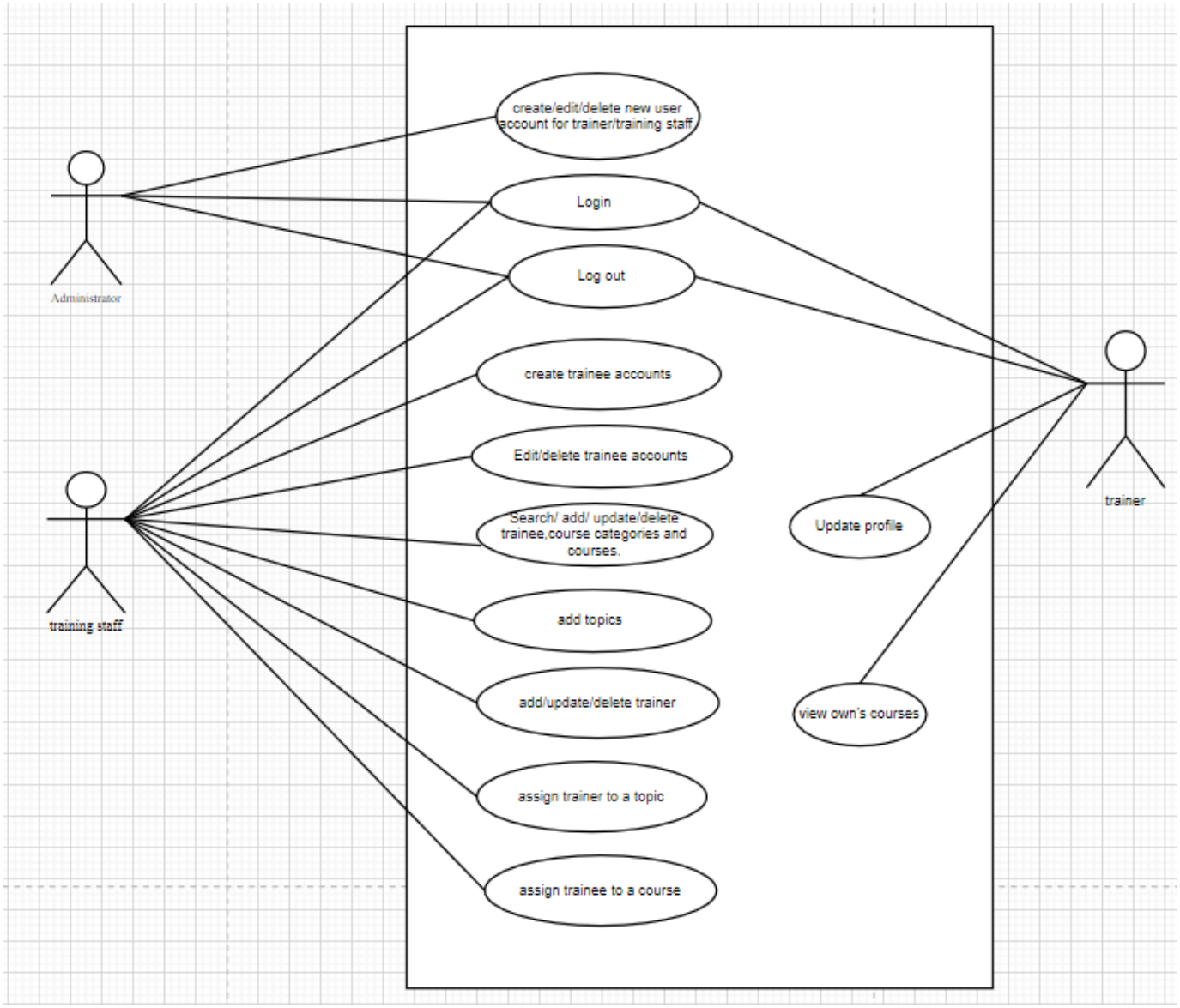
In addition, the second factor maybe is the application that is used to write code and execute code. If the application can't be used with whatever the reasons, the project will be affected.

# III. Specific Requirements

## 1. ERD




## 2. Use Case




# IV. Other Non-functional Requirements

## 1. Performance Requirements

Based on our interview with about 10 people in the company, there are some performance requirements for the the system:

Firstly, the system must have high website loading speed ( maximum is 1 second/loading)

Secondly, about security, the system must have high security ( especially in the database). If the database is hacked, hackers can't take data from it.

Thirdly, the system can perform multiple access at the same time without delay.

Finally, whenever users login into the system. Never have lag phenomena or delay between the loading page.

## 2. Safety and Security Requirements

Based on our interview with about 10 people in the company, there are some requirements that are concerned with possible loss, damage, or harm that could result from the use of the product.

About the security problem, need to prevent hackers attack to the database. Moreover, should have some safety certifications, such as:

- Associate Safety Professional (ASP)
- Graduate Safety Practitioner (GSP)
- Certified Safety Professional (CSP)

Besides, need to secure the password of all accounts in the database into standard MD5, prevent hackers take data then sell them. In addition, should have more ways to help secure the system.

## 3. Software Quality Attributes

### a. Flexibility

The system needs flexibility in web design. There are now more devices and web browsers than ever before and they all need to be catered for. All system designs need to be flexible, adaptable and be suitable for multiple formats. More than that, they need to look good on different devices. So need to create a design that was suitable for all sizes of screens and all devices.

### b. Testability

In addition to flexibility, the system needs to have testability. Because whenever the system has good testability, the system can be maintained easier. Moreover, it helps developers and testers do the tasks conveniently

For example, one of ways can be used is unique identifiers.

Especially when doing UI testing, it’s important to have unique identifiers on the page. There are different approaches: assign an `id` to every element or assign a unique identifier to each component and input element. Implementing id’s for every element is costly and unnecessary. Implementing unique identifiers for every component is a better approach because they can search relatively within that component. Like, you have a search component with an input field and button:

```
<section id="search">
 <input type="text" name="query">
 <button type="submit" name="search">Buscar</button>
<section>
```

## V. Appendix A – Data Dictionary

| Table | Name | Data Type | Null | Primary/ Foreign Key |
|---|---|---|---|---|
| **Admin** | Admin_id | string | Not null | PK |
| | Admin_phone | string | | |
| | Admin_email | string | | |
| | Staff_id | string | Not null | FK |
| | Trainer_id | string | Not null | FK |
| | Account_id | string | Not null | FK |
| | | | | |
| **Training Staff** | Staff_id | string | Not null | PK |
| | Staff_name | string | Not null | |
| | Staff_phone | string | | |
| | Staff_email | string | | |
| | Category_id | string | Not null | FK |
| | Trainer_id | string | Not null | FK |
| | Trainee_id | string | Not null | FK |
| | Account_id | string | Not null | FK |
| | | | | |
| **Trainer** | Trainer_id | String | Not null | PK |
| | Trainer_name | String | Not null | |
| | trainer_phone | String | | |
| | Trainer_email | String | | |
| | Working_place | String | | |
| | Course | String | | |
| | type | String | | |
| | Account_id | String | Not null | FK |
| | | | | |
| **Trainee** | Trainee_id | String | Not null | PK |
| | Trainee_name | String | Not null | |
| | Age | Int | | |
| | DoB | String | | |
| | Education | String | | |
| | Programming_language | String | | |
| | Toeic_score | Int | | |
| | Experience_detail | String | | |
| | Department | String | | |

| | | | | |
|---|---|---|---|---|
| | location | String | | |
| | | | | |
| **Account** | Account_id | String | Not null | PK |
| | Account_name | String | Not null | |
| | Account_pass | String | Not null | |
| | Role_id | int | Not null | FK |
| | | | | |
| **User Role** | Role_id | String | Not null | PK |
| | Role_name | String | Not null | |
| | | | | |
| **Course Category** | Category_id | String | Not null | PK |
| | Course_id | String | Not null | FK |
| | Category_name | String | Not null | |
| | Category_description | String | | |
| | | | | |
| **Course** | Course_id | String | Not null | PK |
| | Topic_id | String | Not null | FK |
| | Course_name | String | Not null | |
| | Course_description | String | | |
| | | | | |
| **Topic** | Topic_id | String | Not null | PK |
| | Topic_name | String | Not null | |
| | Topic_description | String | | |

## VI. Appendix B - Group Log

| Time | Activities |
|---|---|
| **12 / 09** | Make clear requirements |
| **17/09** | Continue making clear requirements |
| **19 / 09** | Draw ERD and use case diagram |
| **24 / 09** | Divide work for each member |
| **26 / 09** | Continue completing work |
| **01 / 10** | Check work of each member |
| **03 / 10** | Complete slide for presentation |
| **08 / 10** | Complete report |