UNIVERSITY of GREENWICH
Alliance with FPT Education

BTEC

Higher Nationals in Computing

UNIT 30

UNIT 30: APPLICATION DEVELOPMENT

ASSIGNMENT No.1

Learner's name: Trinh Thi Dieu Huyen

ID: GDD18606

Class: GCS0801_NX

Subject code: 1670

Assessor name: **THAI THI THANH THAO**

Assignment due: April 2021

Assignment submitted: April 2021

ASSIGNMENT 1 FRONT SHEET

Qualification:	BTEC HND Diploma in Computing and Systems Development		
Unit number and title:	Unit 30: Application Development		
Assignment due:	April 2021	**Assignment submitted:**	April 2021
Learner's name:	Trinh Thi Dieu Huyen	**Assessor name:**	Thai Thi Thanh Thao
Learner declaration:	I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.		
Learner signature:	huyen	**Date:**	April 2021

Grading grid

P1	P2	P3	M1	M2	D1

Assignment title	**Assignment 1: Requirement Analysis and Application Design**
In this assignment, you will have opportunities to provide evidence against the following criteria. Indicate the page numbers where the evidence can be found.	

ASSESSMENT CRITERIA	EXPECTED EVIDENCE	TASK	ASSESSOR'S FEEDBACK
LO1. Produce a Software Design Document by analysing a business-related problem and deduce an appropriate solution including a set of initial requirements			
P1 Explore a business-related problem and produce a well-defined Problem Definition Statement supported by a set of user and system requirements.	1. Use case diagram to capture problem requirements 2. Screen-flow diagram and wireframes (prototype screen)	1	
P2 Determine any areas of risk related to the successful completion of your application.	3. Questions to identity any problem from the requirements, use cases, screen-flow diagram or UI 4. Answer to above questions according to personal research	2	
LO2. Use design and development methodologies with tools and techniques associated with the creation of a business application			
P3 Research the use of software development tools and techniques and identify any that have been selected for the development of this application.	5. ERD describing application database 6. UML Class diagram for the core business processing	3	

Assessment criteria	Expected Evidence	Feedback (note on Merit/Distinction if applicable)
Merit descriptor No. (M1)		
Merit descriptor No. (M2)		
Distinction descriptor No. (D1)		

Summative feedbacks:			
Assessor's Signature:		**Date:**	

ASSIGNMENT 1 BRIEF

Unit Number and Title	**Unit 30: Application Development**
Academic Year	2021
Unit Tutor	Thai Thi Thanh Thao
Assignment Title	**Assignment 1: Requirement Analysis and Application Design**
Issue Date	**April 2021**
Submission Date	April 2021
IV Name & Date	Thai Thi Thanh Thao March 2021

SUBMISSION FORMAT
The submission is in the form of 1 word document containing required evidences and 1 presentation if necessary (M2, D1). You are required to make use of headings, paragraphs, subsections and illustrations as appropriate, and all work must be supported with research and referenced using the Harvard referencing system.

ASSIGNMENT BRIEF AND GUIDANCE
As the technology is being developed rapidly nowadays, FPT Co. desires to build the continuing study environment throughout the corporation. It is necessary to develop a web-based system, which manages the activity of "Training" for internal training program of the company. 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. This is a system used by HR department. We have three roles in this system, an administrator, training staff and a trainer. The brief description of those roles is as follow. **1. An administrator's role** ▪ Can login to the system through the first page of the application ▪ Can create/edit/delete new user account for trainer/training staff and assign/change(if existing user)

username and a password

2. A training staff's role

- A registered training staff, who is assigned a user name and a password by the administrator logs in can create trainee accounts by entering details like trainee name, trainee accounts, age, date of birth, education, main programming language, TOEIC score, experience details, department, location, etc.
- After entering successfully all details for trainees, his/her details are then stored in the database. The training staff is given a list of trainees for him to view and search. From the list of trainees, he can also search by trainee account, programming language, TOEIC score…
- Can update, delete trainee accounts
- Can manage course categories such as searching, adding, updating and deleting course categories. Course category includes the information such as course category name and descriptions.
- Can manage courses such as searching, adding, updating and deleting courses. Course includes course name and description.
- Can add topics such as topic name and topic descriptions into a course, add courses into a category.
- Can manage trainer profile such as adding, updating and deleting the information: Trainer name, External or Internal Type, working place, telephone, and email address.
- Can assign trainer to a topic.
- Can assign trainee to a course.

3. A trainer's role

- In the same system, the trainer who have been registered by the administrator can login and can update his profile such as Trainer name, External or Internal Type, education, working place, telephone, and email address.
- Can view courses which have a topic he is assigned to.

Your manager suggests that this would be a great opportunity for you to demonstrate your capabilities by designing and developing the application. After considering, you decide to do the project. The project consists of 4 steps which is divided into two phases. In this first phase, you have to carry on requirement analysis and produce solution design for the problem.

ASSIGNMENT GUIDANCE		
Task	**Assessment Criteria**	**Requirement**
1	**P1** Explore a business-related problem and produce a well-defined Problem Definition Statement supported by a set of user and system requirements.	○ *Use case diagram to capture problem requirements* ○ *Screen-flow diagram and wireframes (prototype screen)*
2	**P2** Determine any areas of risk related to the successful completion of your application.	○ *Questions to identity any problem from the requirements, use cases, screen-flow diagram or UI* ○ *Answer to above questions according to personal research*
3	**P3** Research the use of software development tools and techniques and identify any that have been selected for the development of this application.	○ *ERD describing application database* ○ *UML Class diagram for the core business processing*
4	**M1** Analyse a business-related problem using appropriate methods and produce a well-structured Software Design Document that defines a proposed solution and includes relevant details on requirements, system analysis, system design, coding, testing and implementation.	○ *Detail use case description for important functions*
5	**M2** Compare the differences between the various software development tools and techniques researched and justify your preferred selection as well as your preferred software development methodology.	○ *A presentation to introduce different software development methods (at least 3), for example: Waterfall, Prototyping, Spiral, Agile…*
6	**D1** Justify the tools and techniques chosen to realise a custom built website. Justify your preferred selection of tools and techniques in deducing an appropriate solution to a business related problem.	○ *Critical comparison among development methods in the presentation and reasons for your selection of tools and techniques.*

<table>
<tr><th colspan="3">Learning Outcomes and Assessment Criteria</th></tr>
<tr><th>Pass</th><th>Merit</th><th>Distinction</th></tr>
<tr><td colspan="2">LO1 Produce a Software Design Document by analysing a business-related problem and deduce an appropriate solution including a set of initial requirements</td><td rowspan="5">LO1 & 2

D1 Justify the tools and techniques chosen to realise a custom built website. Justify your preferred selection of tools and techniques in deducing an appropriate solution to a business related problem.</td></tr>
<tr><td>P1 Explore a business-related problem and produce a well-defined Problem Definition Statement supported by a set of user and system requirements.

P2 Determine any areas of risk related to the successful completion of your application.</td><td>M1 Analyse a business-related problem using appropriate methods and produce a well-structured Software Design Document that defines a proposed solution and includes relevant details on requirements, system analysis, system design, coding, testing and implementation.</td></tr>
<tr><td colspan="2">LO2 Use design and development methodologies with tools and techniques associated with the creation of a business application</td></tr>
<tr><td>P3 Research the use of software development tools and techniques and identify any that have been selected for the development of this application.</td><td>M2 Compare the differences between the various software development tools and techniques researched and justify your preferred selection as well as your preferred software development methodology.</td></tr>
</table>

Table of Contents

Table of Figures

Table of Tables

ASSIGNMENT 1 ANSWERS

LO1 Produce a Software Design Document by analyzing a business-related problem and deduce an appropriate solution including a set of initial requirements

P1 Explore a business-related problem and produce a well-defined Problem Definition Statement supported by a set of user and system requirements

1. **Customer requirements**

   This is a system used by the human resource (HR) department and has four roles in this system are the administrator, the training staff, the trainer and the trainee. We use a use-case diagram to capture problem requirements. Besides, we also use a screen-flow diagram and wireframes to show the prototype screen.

2. **UML Use Case Diagram**

   Use Case Diagram captures the system's functionality and requirements by using actors and use cases. Use Cases model the services, tasks, functions that a system needs to perform. Use cases represent high-level functionalities and how a user will handle the system. Use-cases are the core concepts of Unified Modelling language modeling. **[1]**

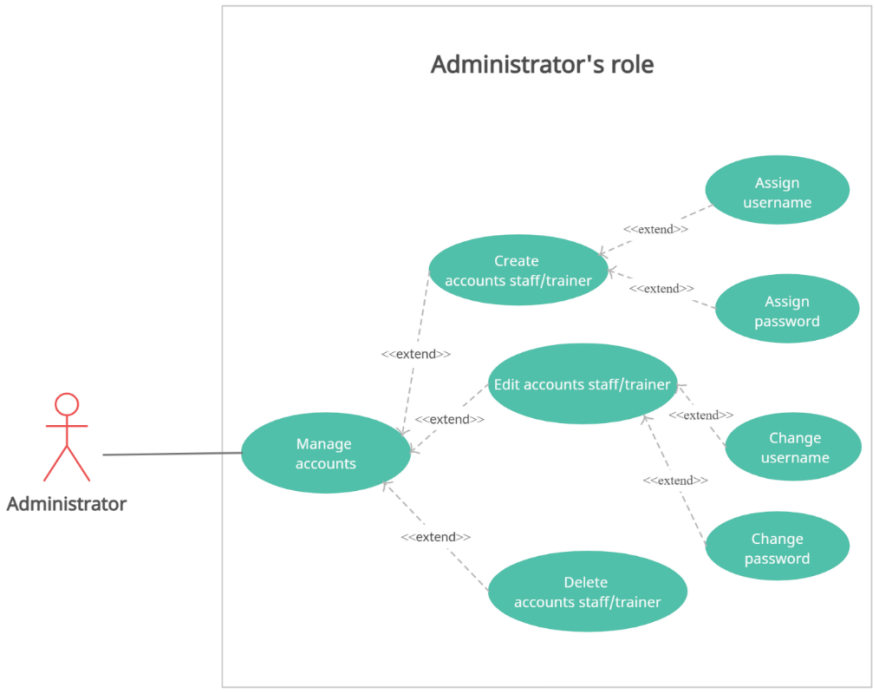

Figure 1: Admin use-case diagram

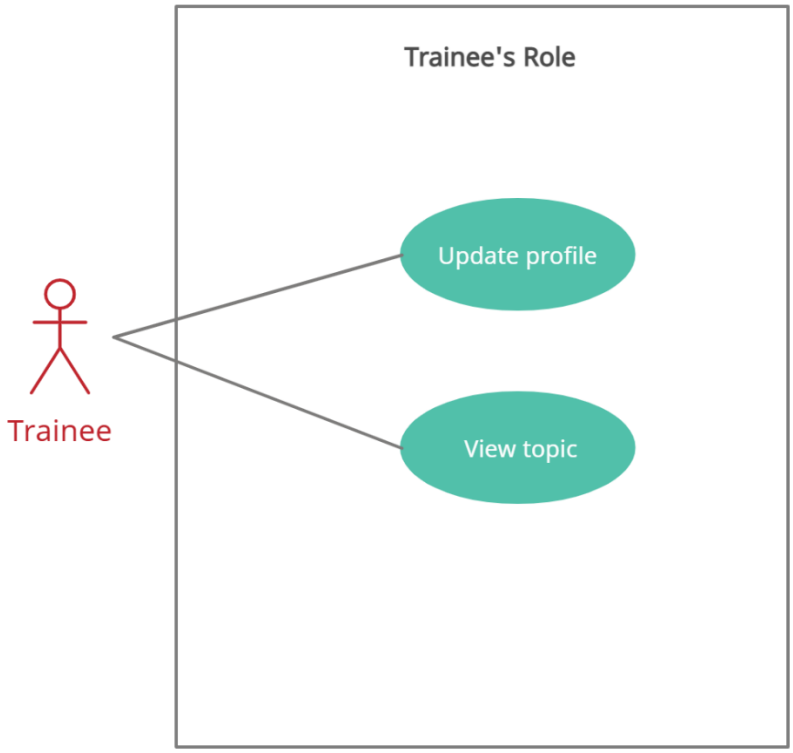

Figure 2: Trainee use-case diagram

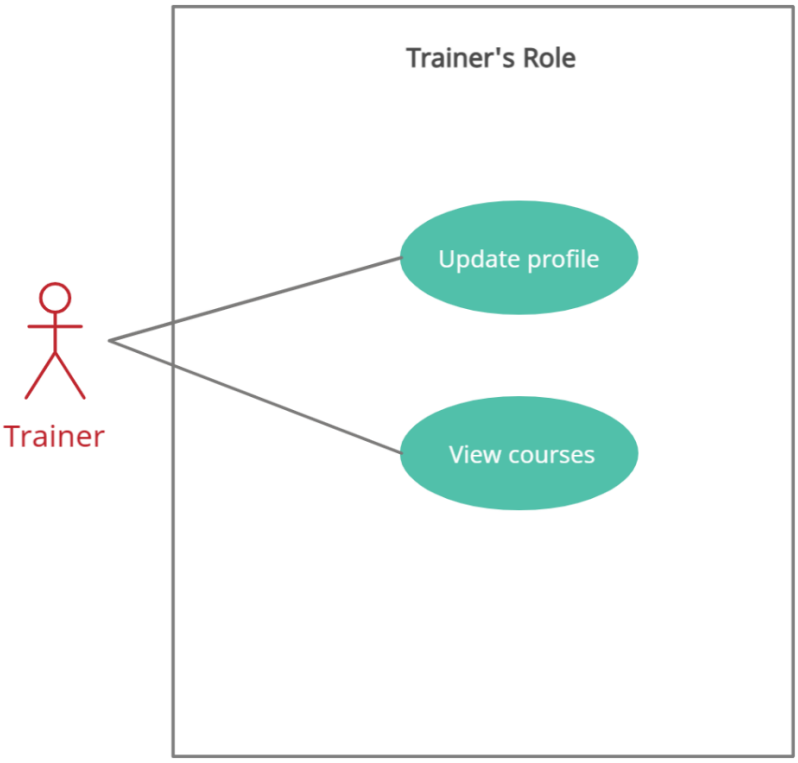

Figure 3: Trainer use-case diagram

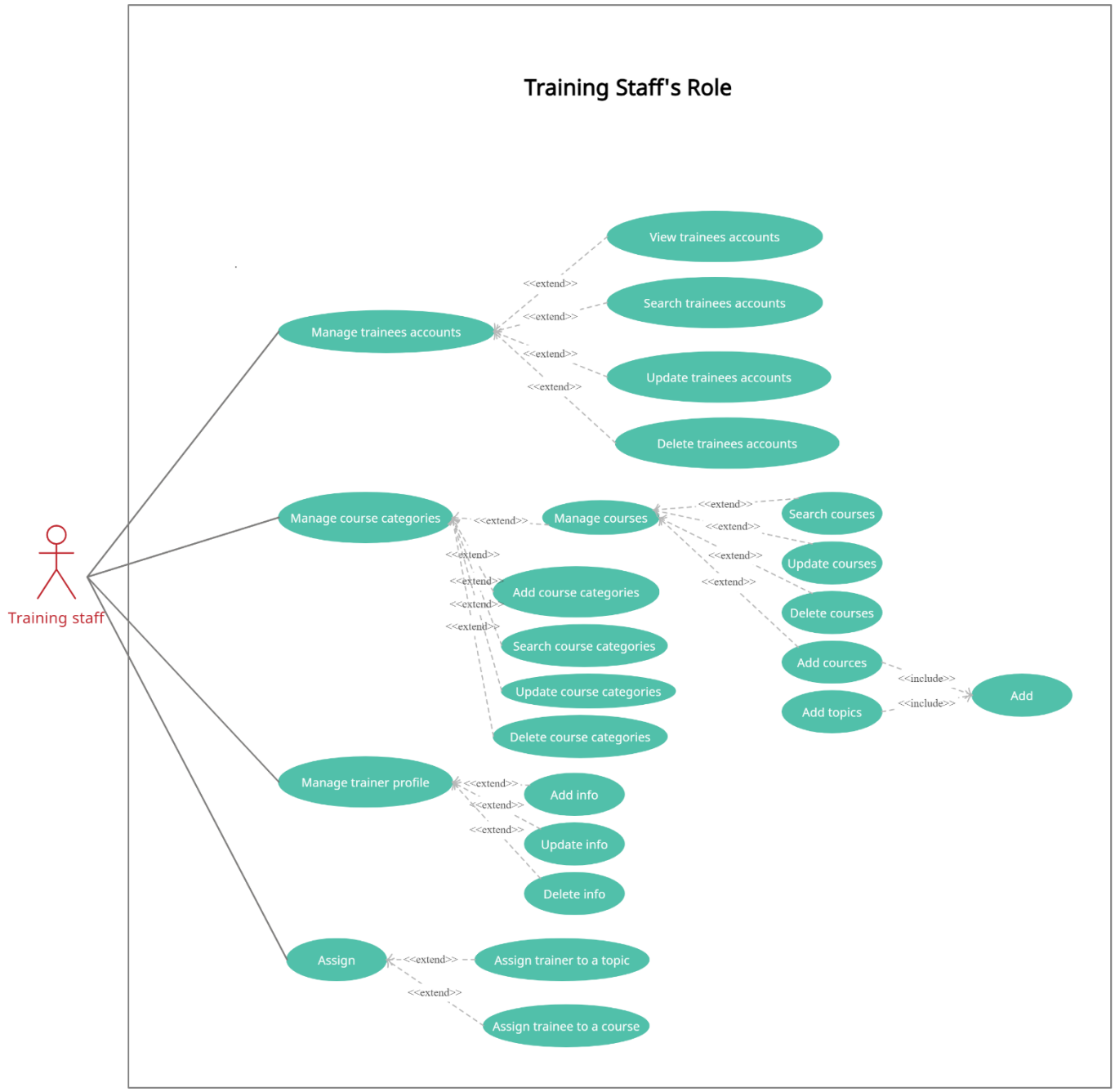

Figure 4: Staff use-case diagram

3. Screen-flow Diagram and Wireframes

3.1. Screen-flow Diagram

Screen-flow diagrams also called storyboards or interface-flow diagrams are windows navigation diagrams, and context-navigation maps. It enables us to model the high-level relationships between major user interface elements and from that ask fundamental usability questions. **[2]**

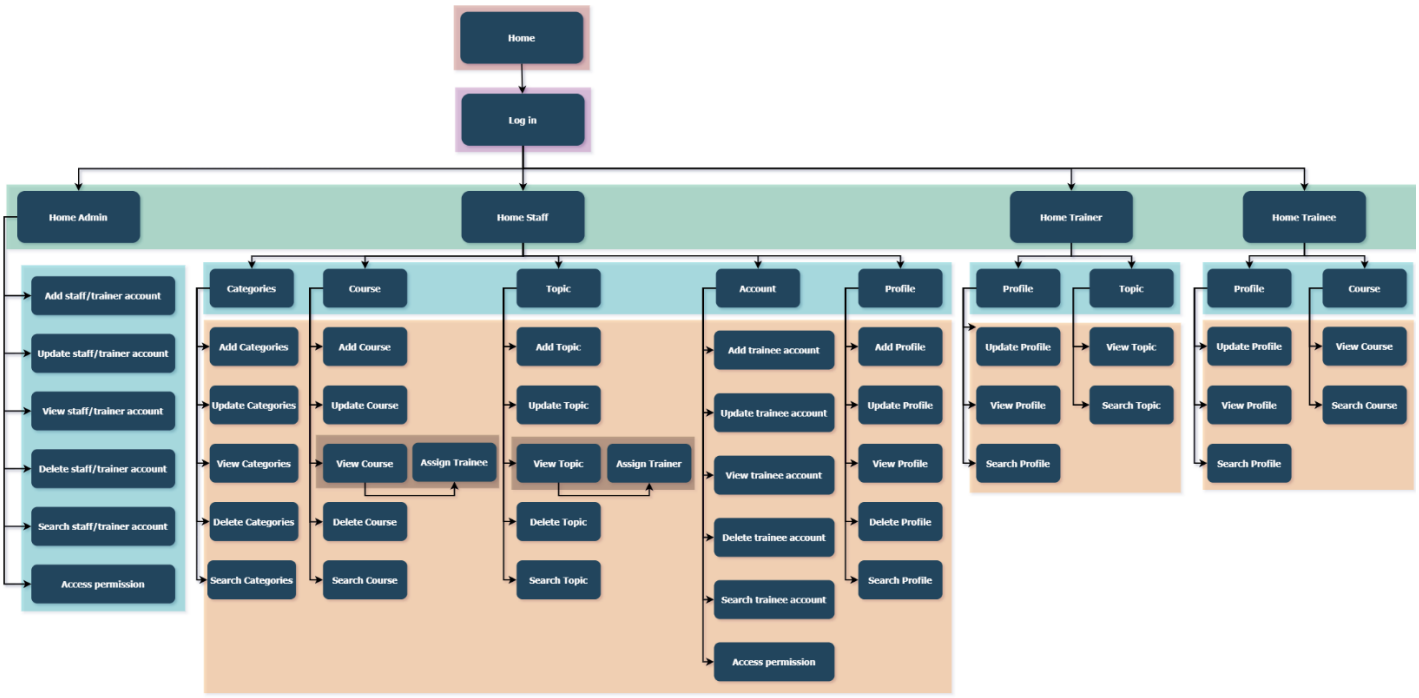

Figure 5: Screen-flow diagram

3.2. Wireframes

The wireframe is a way to design a website service at the structural level. A wireframe is commonly used to layout content and functionality on a page which takes into account user needs and user journeys. Wireframes are used early in the development process to establish the basic structure of a page before visual design and content are added. **[3]**

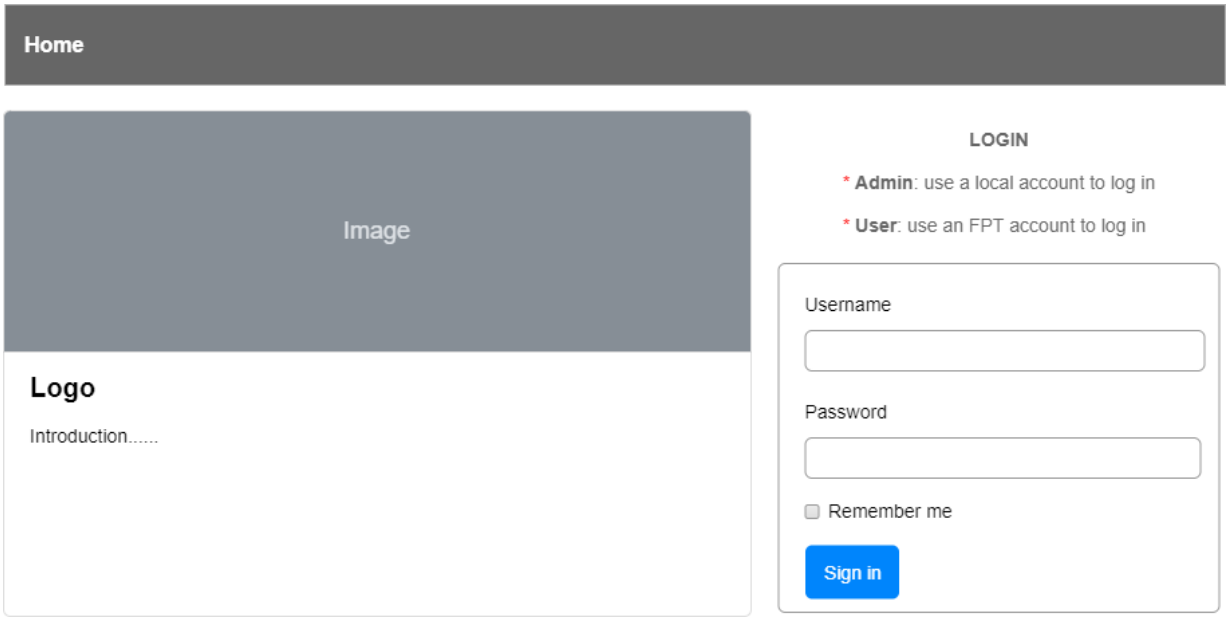

Figure 6: Wireframe for Home/Login

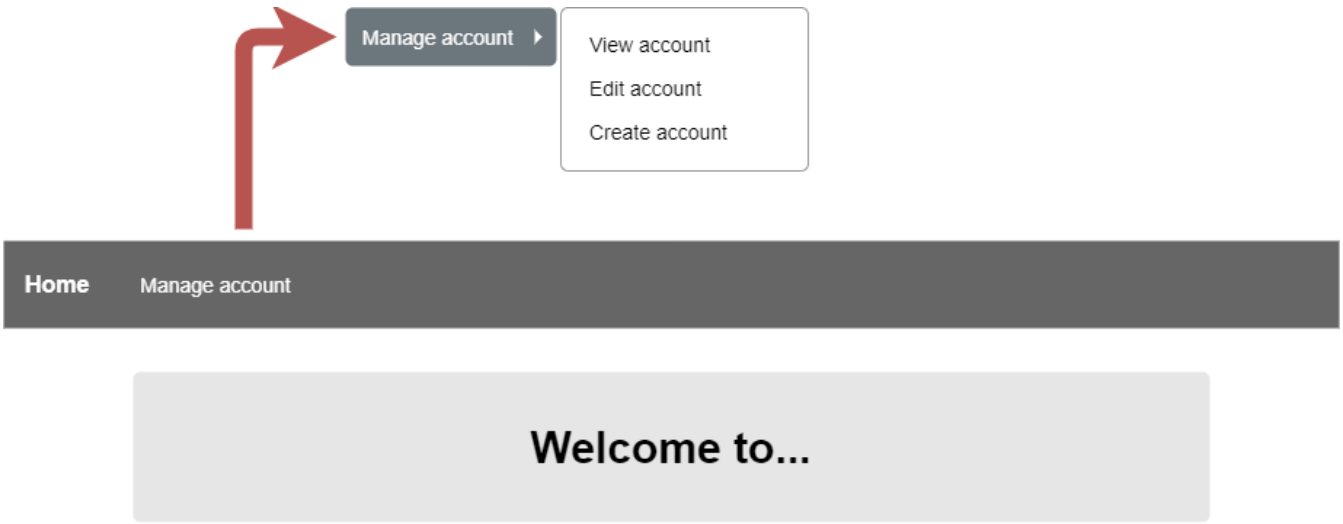

Figure 7: Wireframe for Admin Home

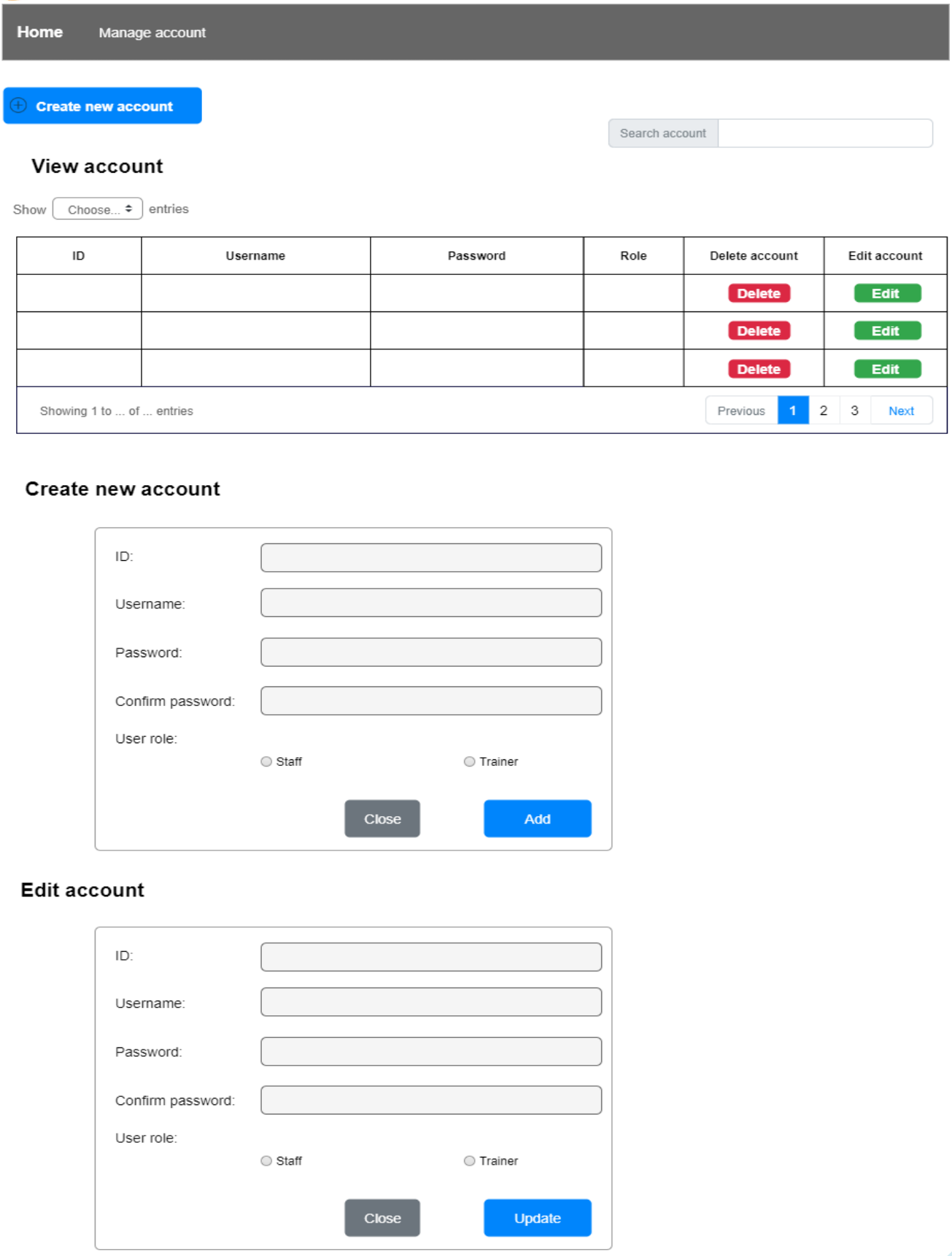

Figure 8: Wireframe for Admin to Manage Staff Account and Trainer Account

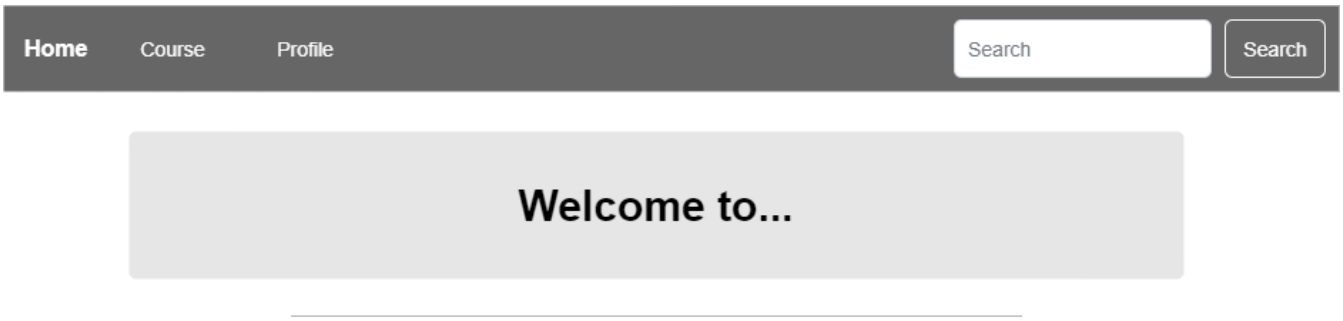

Figure 9: Wireframe for Trainee Home

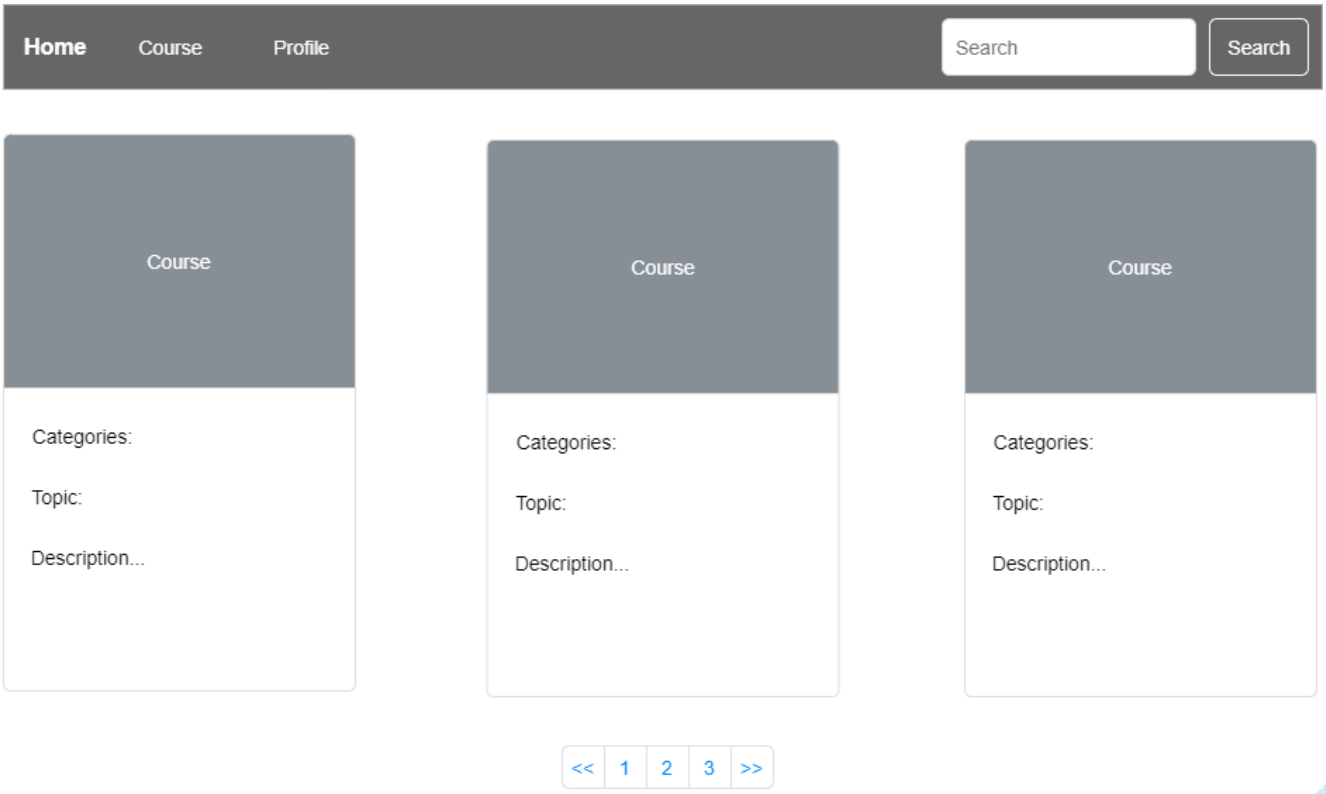

Figure 10: Wireframe for Trainee Course

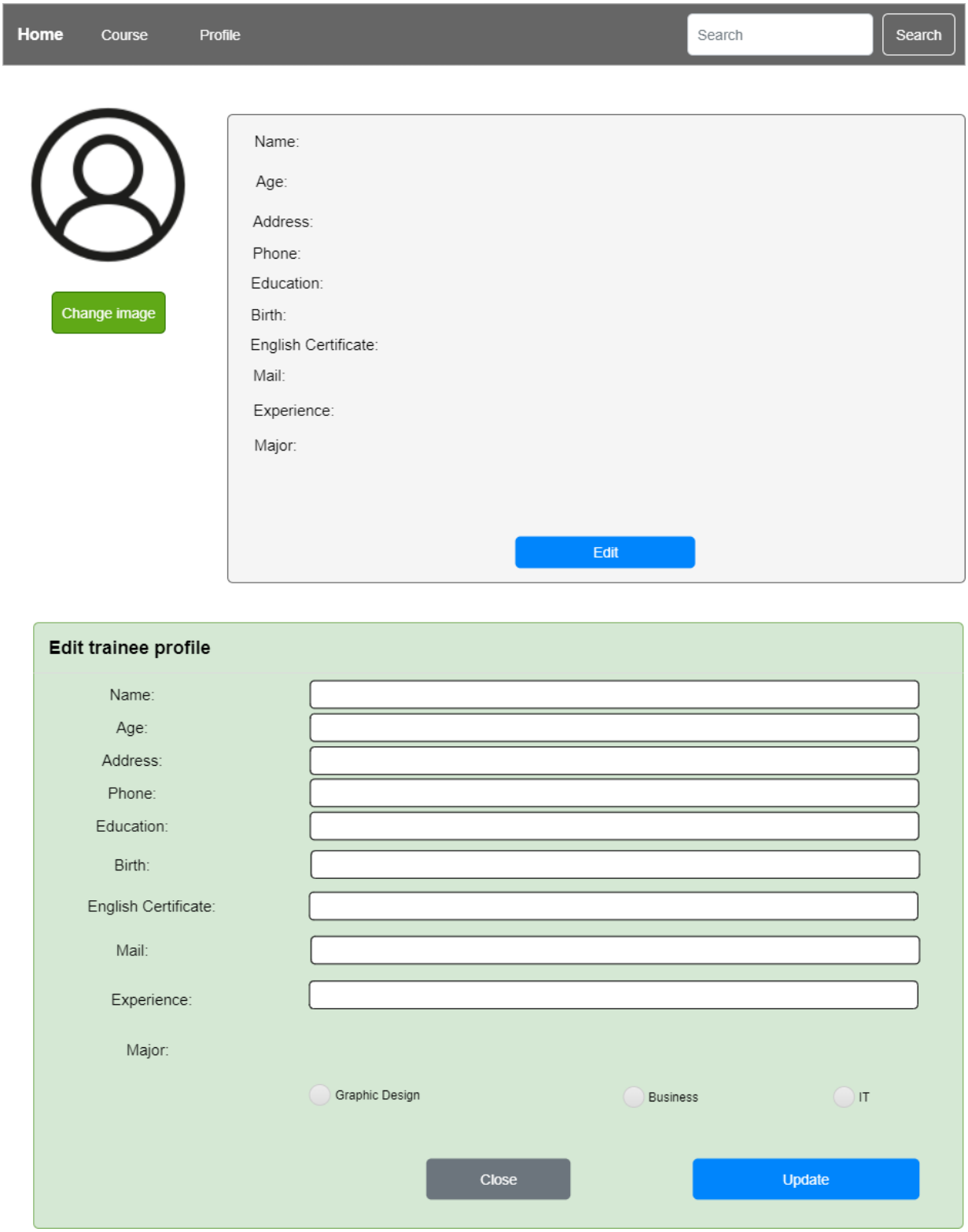

Figure 11: Wireframe for Trainee Profile

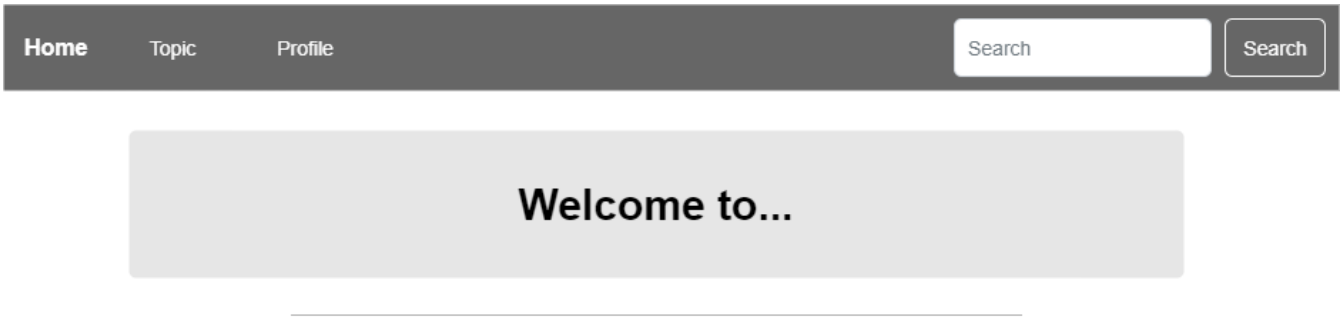

Figure 12: Wireframe for Trainer Home

Figure 13: Wireframe for Trainer Topic

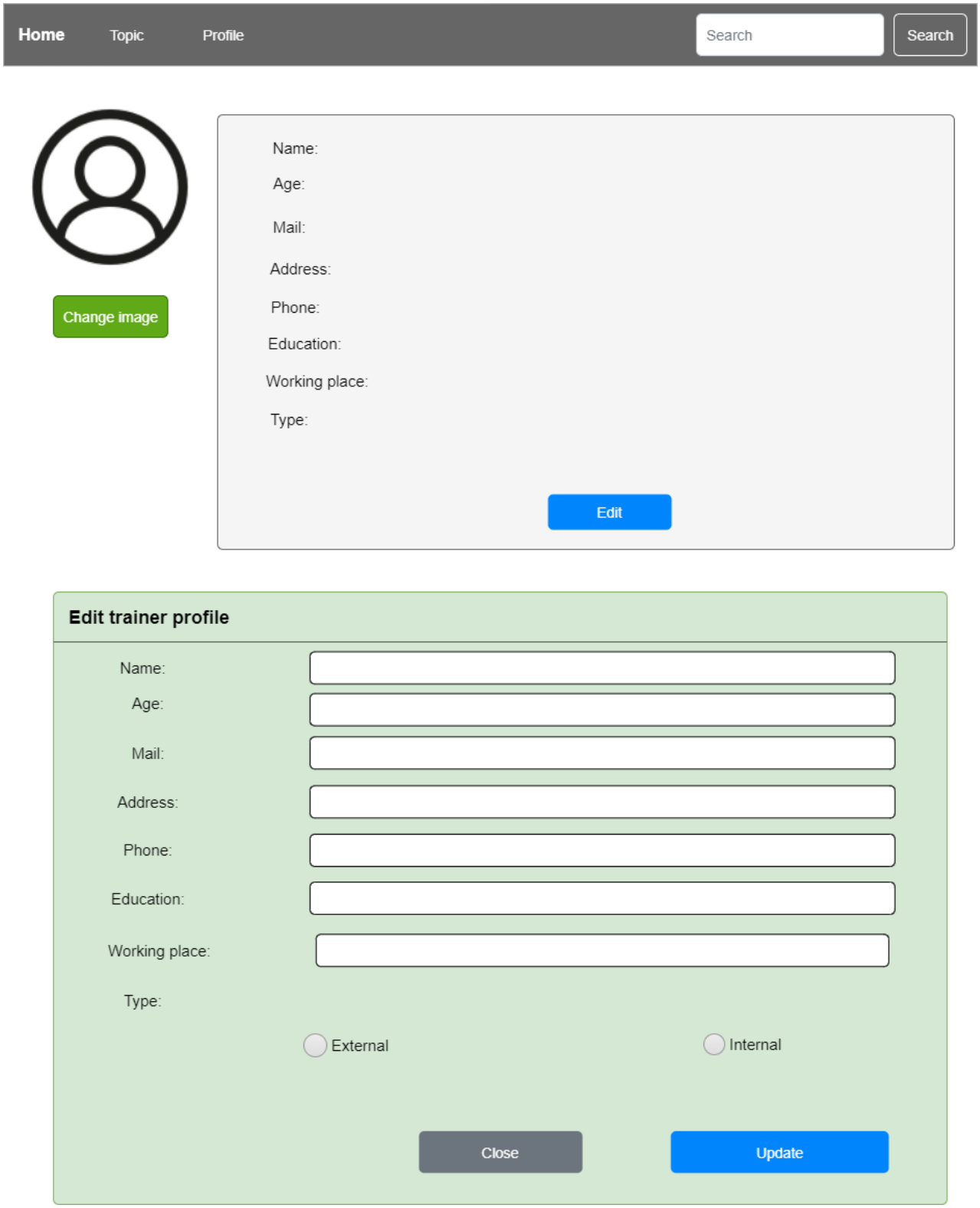

Figure 14: Wireframe for Trainer Profile

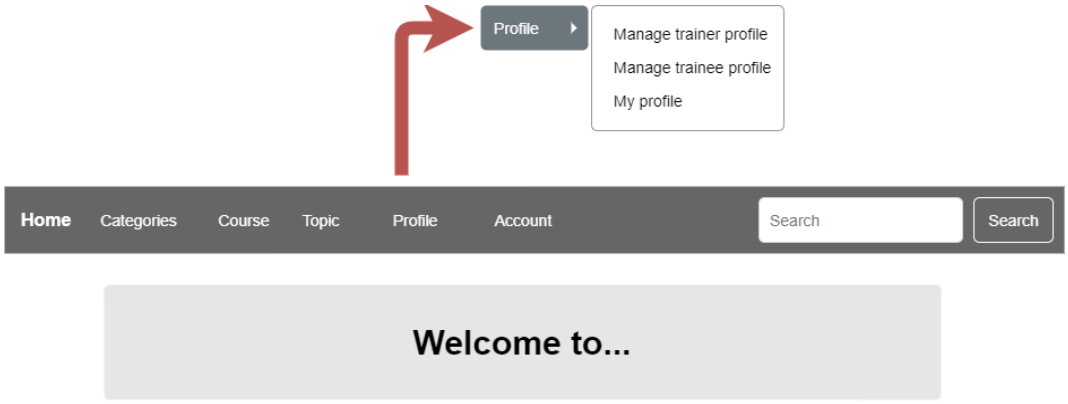

Figure 15: Wireframe for Staff Home

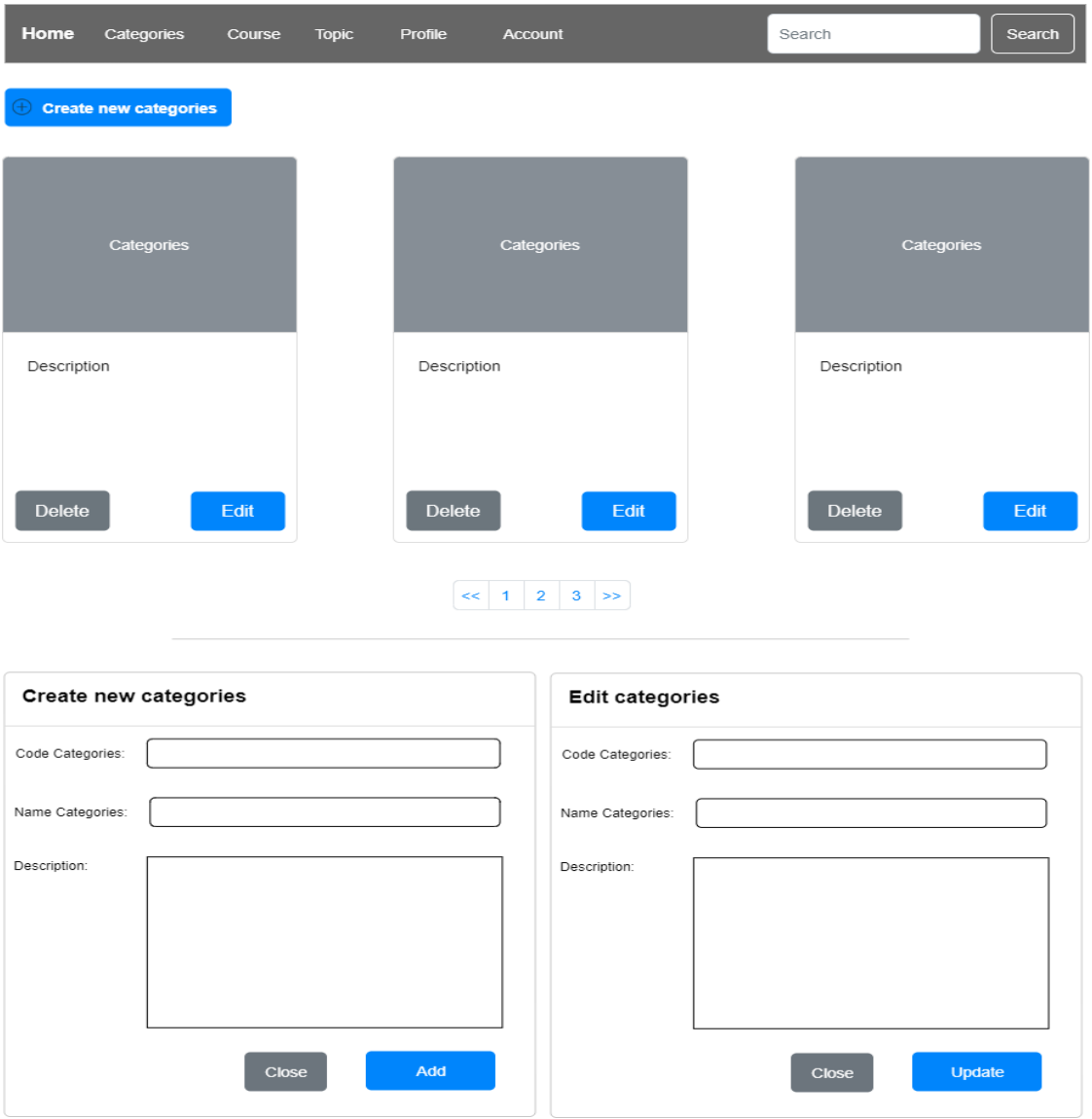

Figure 16: Wireframe for Staff Categories

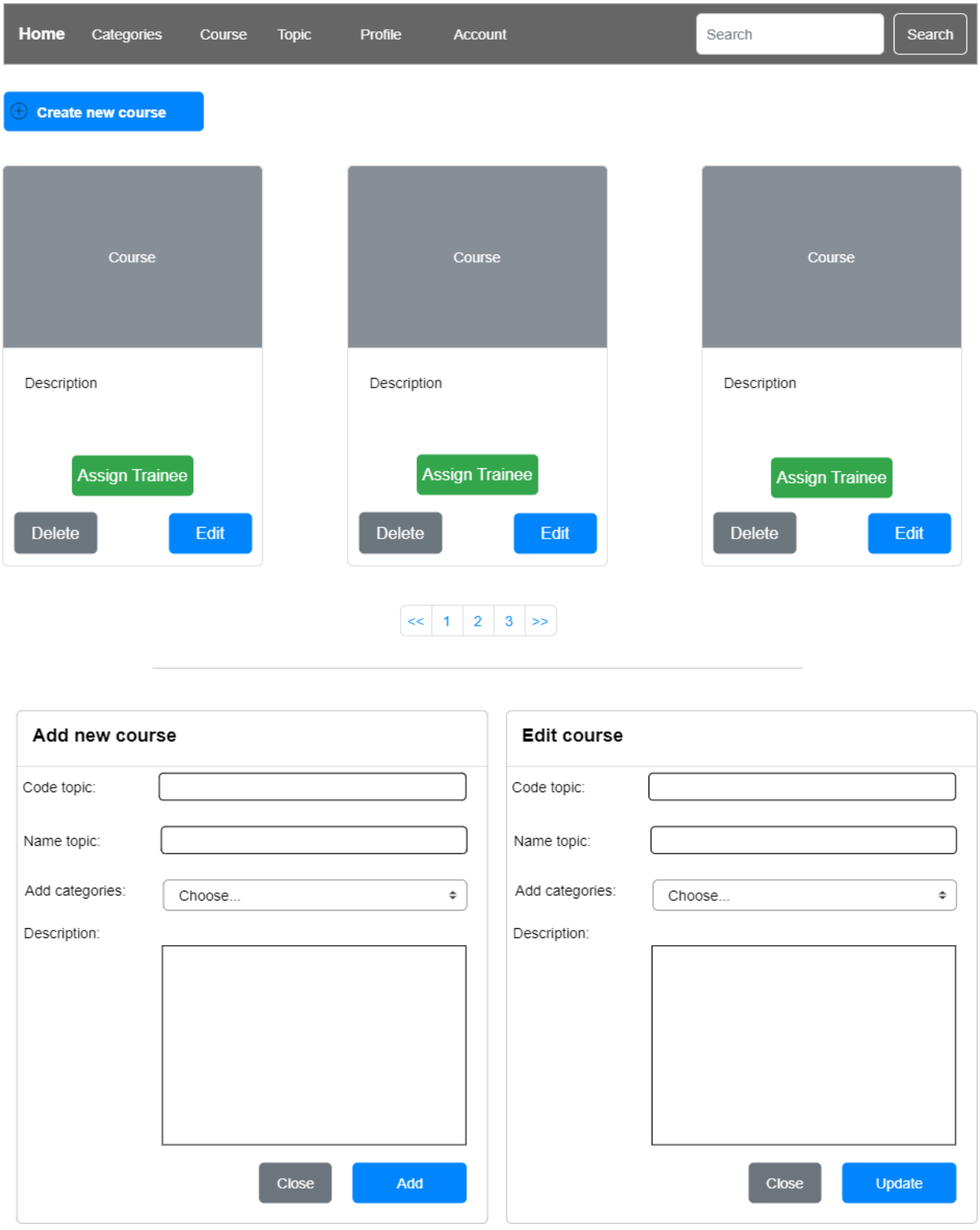

Figure 17: Wireframe for Staff Course

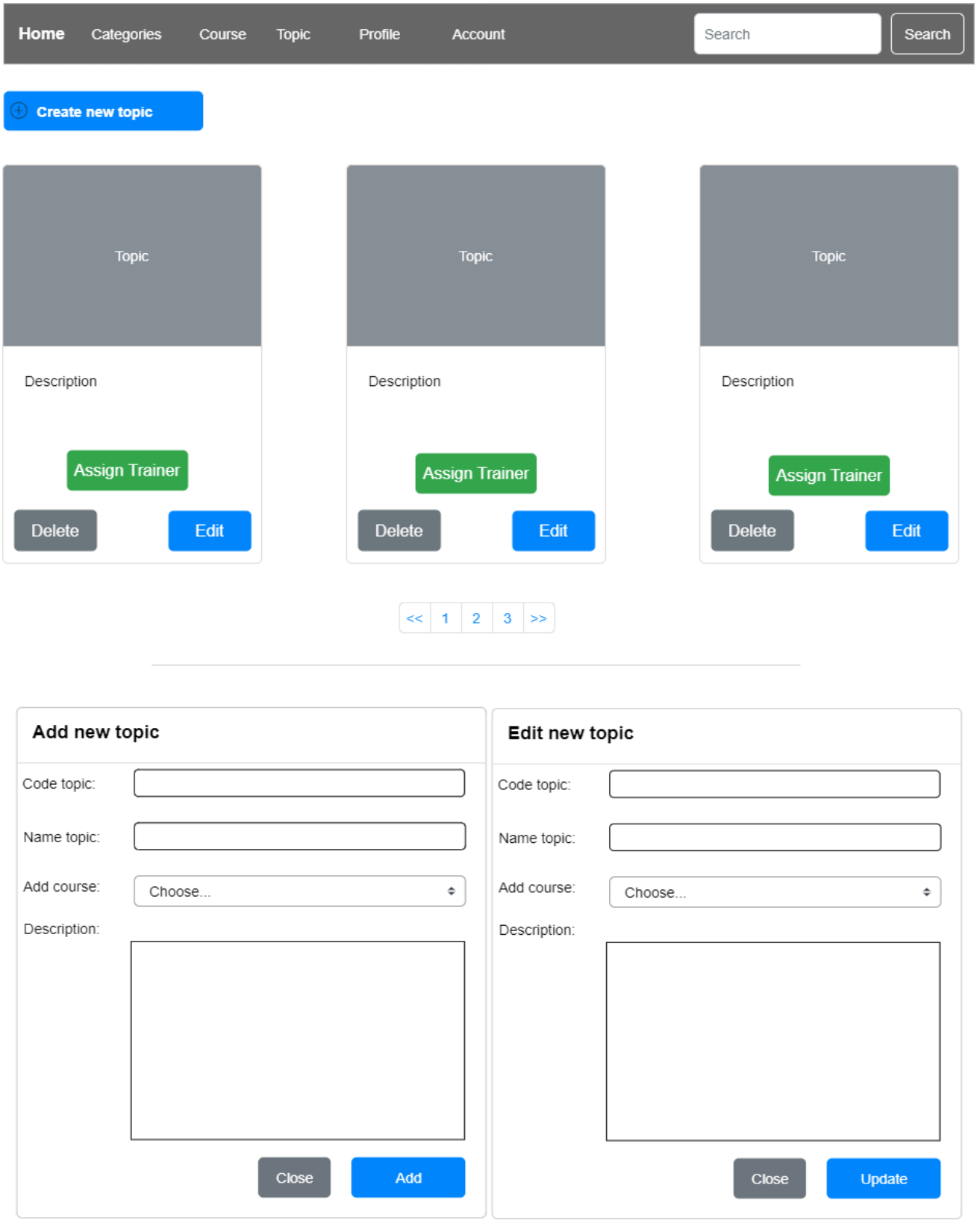

Figure 18: Wireframe for Staff Topic

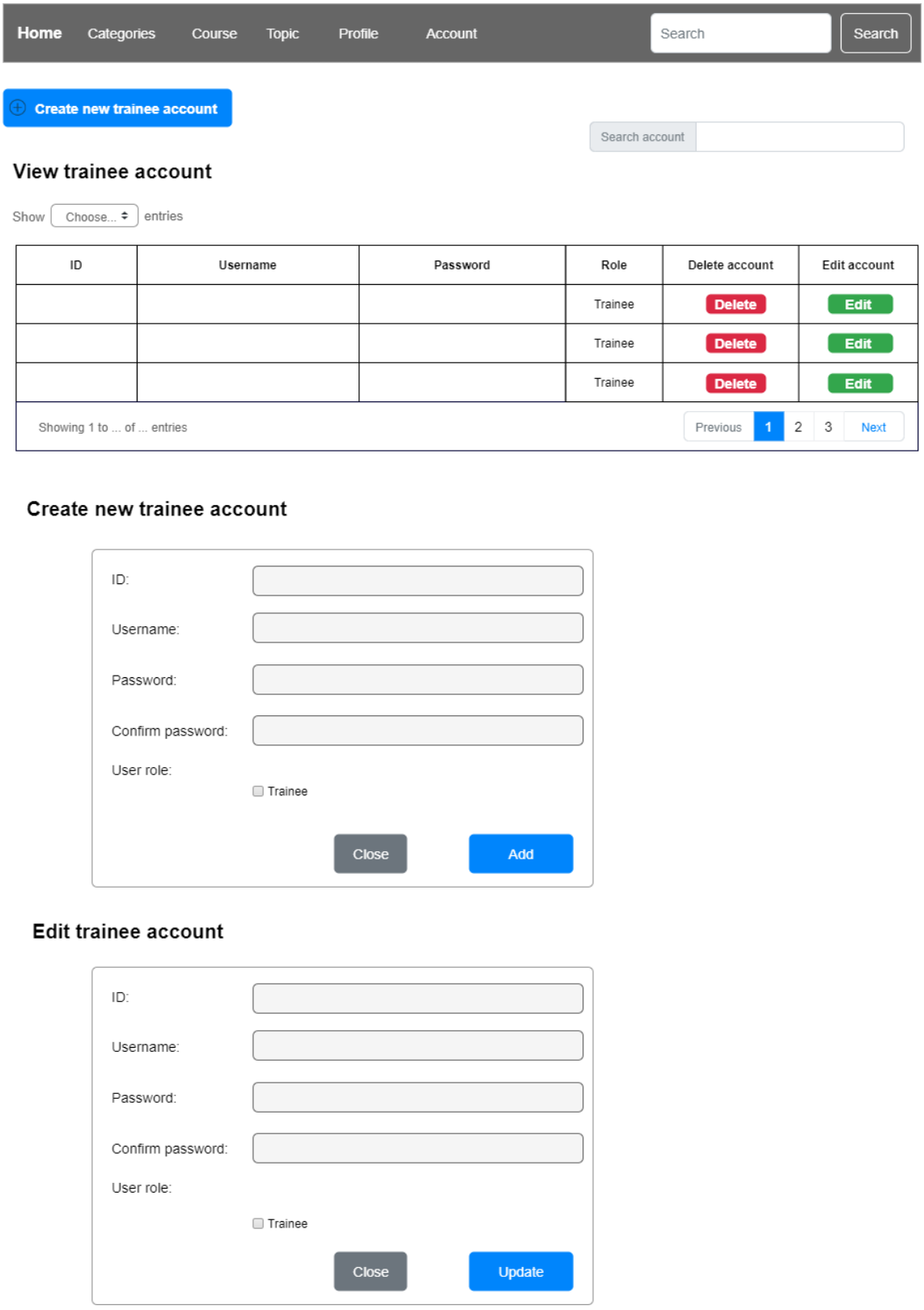

Figure 19: Wireframe for Staff Manage Trainee Account

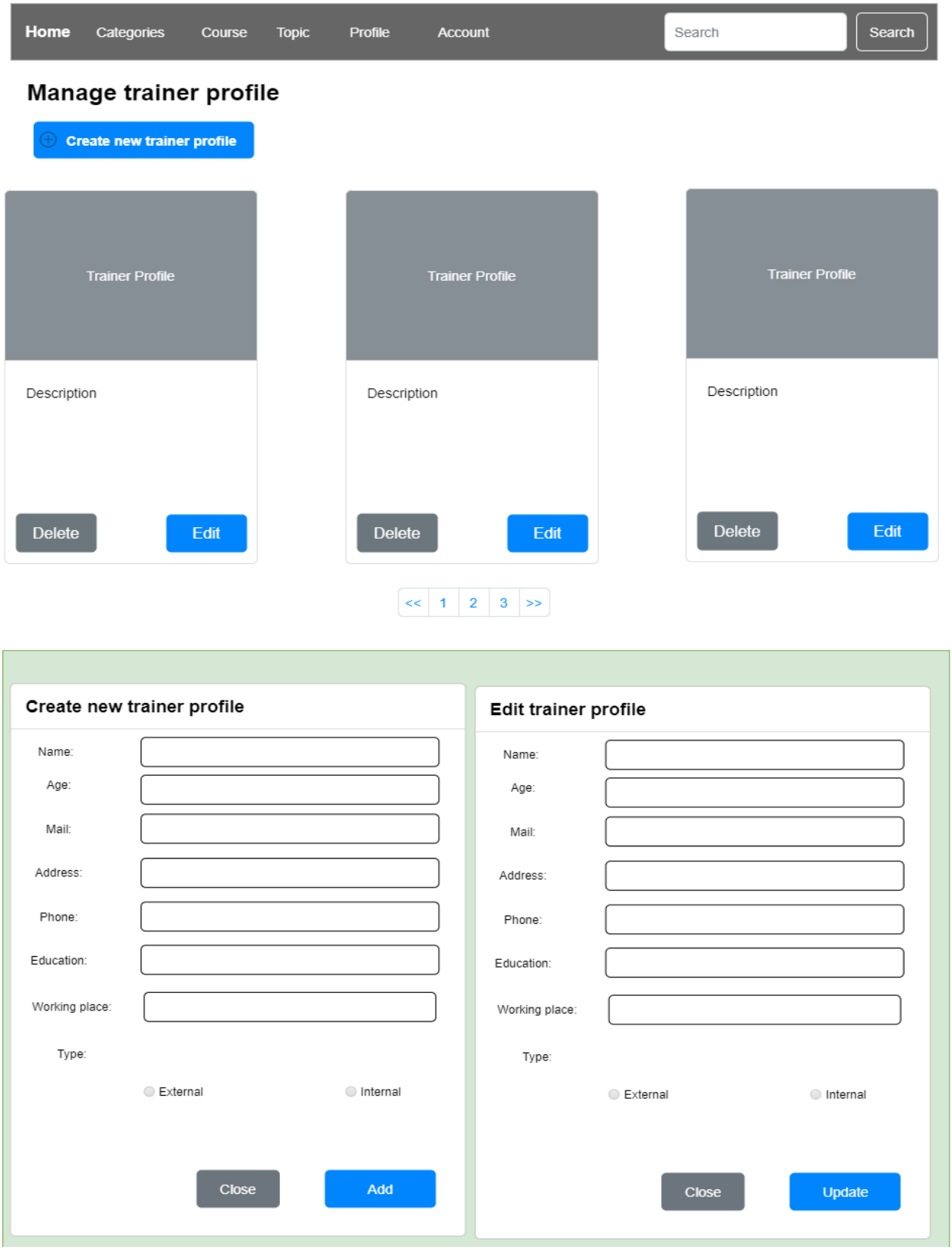

Figure 20: Wireframe for Staff Manage Profile: Trainer

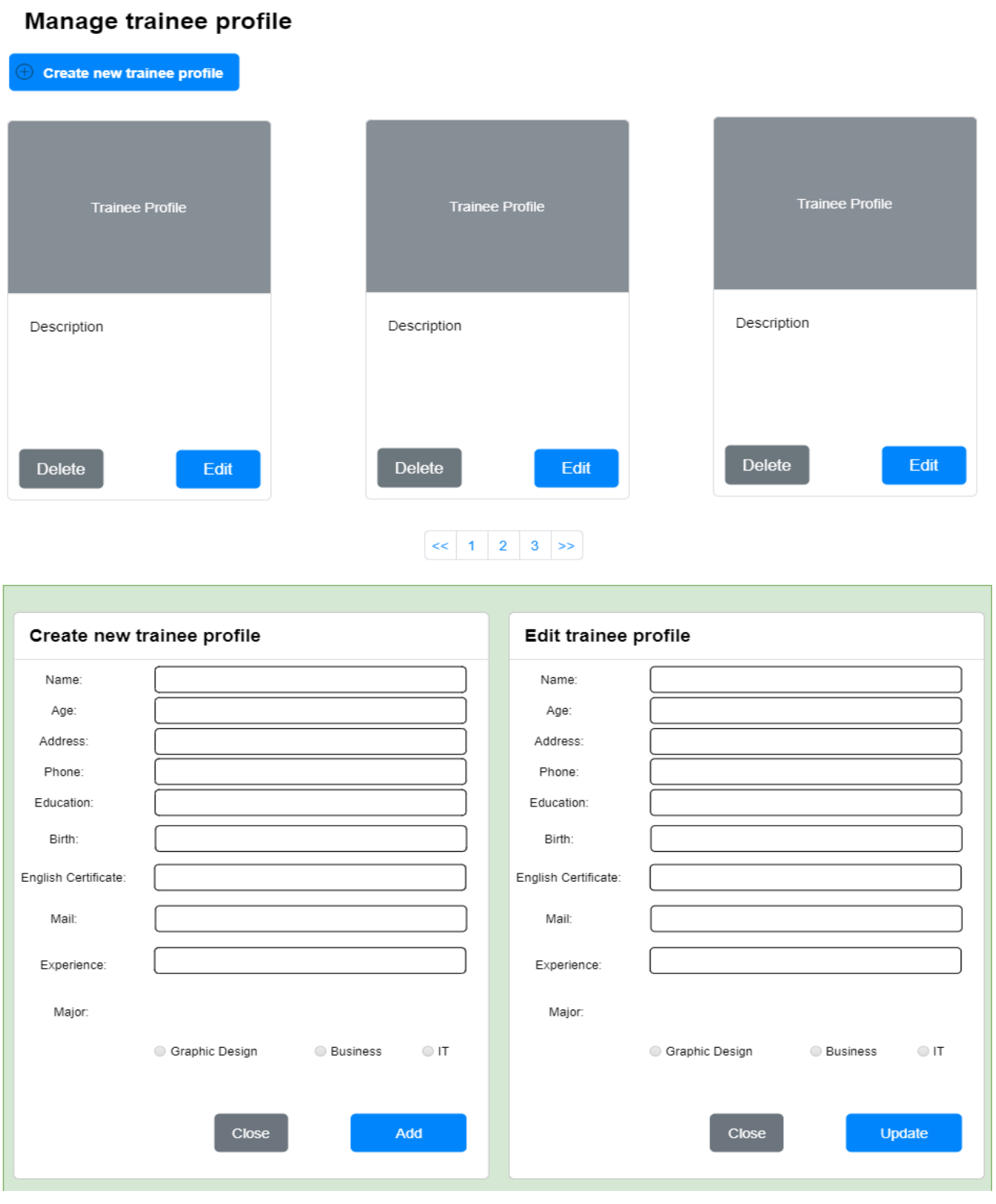

Figure 21: Wireframe for Staff Manage Profile: Trainee

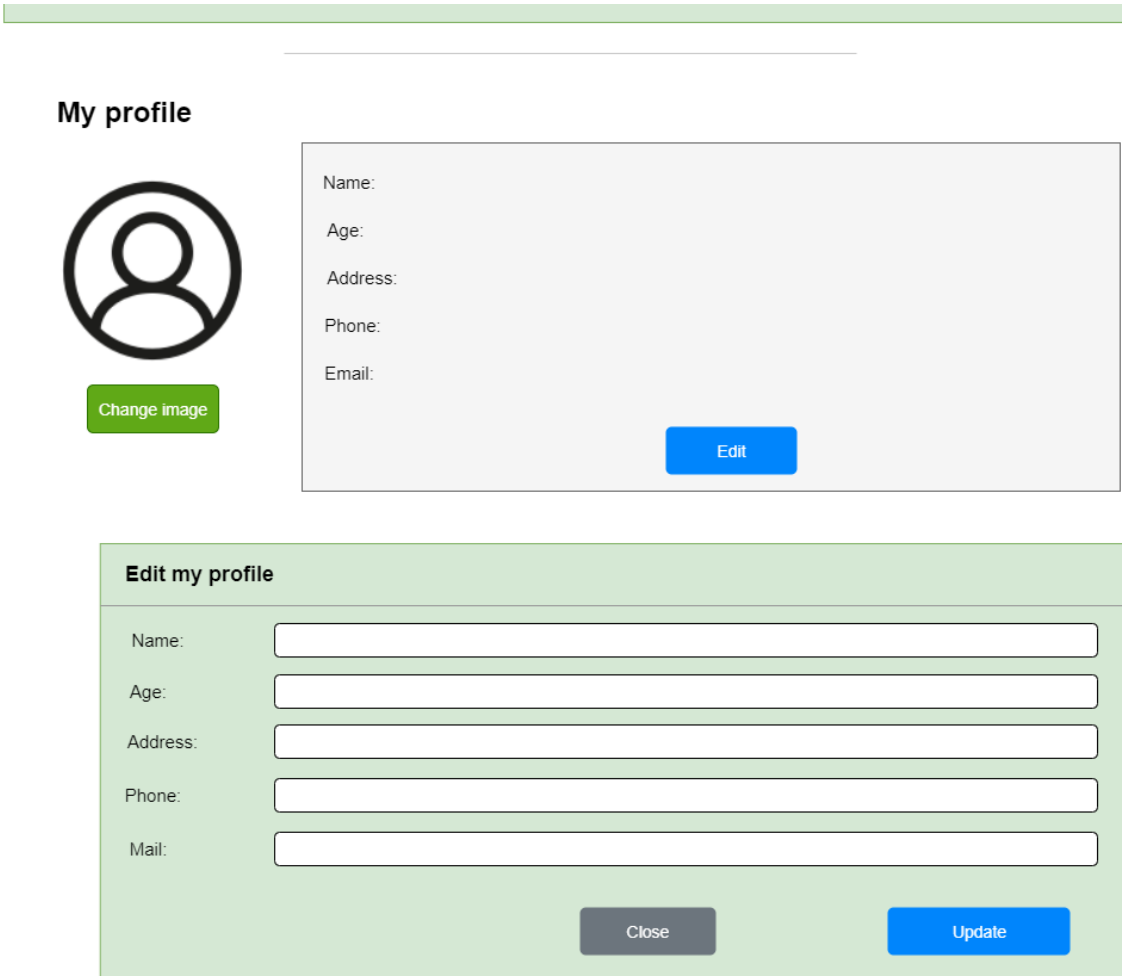

Figure 22: Wireframe for Staff Manage Profile: My Profile (Staff Profile)

P2 Determine any areas of risk related to the successful completion of your application

Risk management is the process of identifying, analyzing, measuring, and assessing risks, from which to find measures to control and overcome the consequences of risks for business activities for optimal use of resources. It helps to organize the implementation of mission goals, business strategies and stable operations. In addition, it also increases the position and reputation of the business or the administrator helps us make the right decisions. On the other hand, it increases safety in the organization's activities and helps businesses to successfully conduct business activities. **[6]**

Below is a table list of some risks that can affect the successful completion of our application.

Table 1: Risk

Name of risk	***Priority***	***Actions***
Lack of human source	Low	Create table division of detail work. Then, jobs that have not been processed due to a shortage of employees are divided among existing employees. Those employees must be paid when doing add that jobs.
Working environment be limited	Low	Improve and decorate the workspace so that the working the environment is no longer issue important
Poor Productivity	Low	Create detailed plans and assign specific jobs to each employee. Divide the work into tasks then assign them to employees who can undertake a piece of work on their own.
Tool is broken (hardware & software)	Medium	Test and maintenance before use
Specification Breakdown	Medium	Use a dedicated Product Manager to make critical trade-off decisions. Test and evaluate before using or making decisions.
Request to change target too many times	Medium	Must stick to the requirement. Re-check the target up to three times before starting to proceed.
Employee turnover	Medium	Signed a commitment not to rotate work when the project is not completed
The sudden growth in requirement	Medium	Set a time for each request increment (1 month/1 request) and limit the number of requirement per request (8-10 requirement/ 1 request)
Inherent Schedule Flaws	High	Get the team more involved in planning and estimating. Get early feedback and address slips directly with stakeholders.
Lack of experience in Application Programming	High	Get advice from experienced people. Refer to the knowledge about Application Programming. Things to keep in mind and what should / shouldn't be for successful Application Programming.

M1 Analyse a business-related problem using appropriate methods and produce a well-structured Software Design Document that defines a proposed solution and includes relevant details on requirements, system analysis, system design, coding, testing and implementation

Table 2: Use case description for login functions

Name	Login	**Code**	#
Description	As admin/user, I want to login to use system		
Actor	Administrator, User (staff, trainer, trainee)	**Trigger**	Admin/user want to login to the system
Pre-condition	- User account has been created - User account has been authorized - User's device is already connected to the internet when perform login		
Post-condition	- The user has successfully logged in to the application - The system records successful login activity		
Standard flow/process	1. Admin/user access to the system 2. Enter your username and password 3. Click login 4. The system verifies successful and allows users to access the application 5. The system records successful login activity Activity Log		
Alternative flow/process	1. Cannot login to the system 2. Show notification 3. Return to home page		
Error situations	1. Cannot connect to the server 2. Account does not exist 3. Account has not been granted access 4. Password or username entered is incorrect 5. Server crash		
System state in error situations	- Still can't login - System down		

Table 3: Use case description for manage account functions

Name	Manage account	**Code**	ID
Description	Admin manage account (staff, tranier). Staff manage account (trainee)		
Actor	Administrator, Staff	**Trigger**	Enter manage account
Pre-condition	Admin/Staff must login in system		
Post-condition	Switch to manage account page		
Standard flow/process	1. Login in system 2. Click Manage account 3. Select the function you want to use (add, edit, search, delete, view) 4. Go to the page containing the selected function 5. Add/edit/search/delete/view account		
Alternative flow/process	1a. Cannot login to the system 2a. Show notification 3a. Return to home page		
	1b. Cannot find this account 2b. Show notification 3b. Return to manage account		
Error situations	1. Cannot connect to the server 2. Account does not exist 3. Account has not been granted access 4. Password or username entered is incorrect 5. Server crash 6. Account does not exist		
System state in error situations	- Still can't login - System down		

Table 4: Use case description for manage categories functions

Name	Manage categories	**Code**	CatID
Description	Staff manage categories: Create, edit, delete, view, search		
Actor	Staff	**Trigger**	Enter manage categories
Pre-condition	Staff must login in system		
Post-condition	Switch to manage categories page		
Standard flow/process	1. Login in system 2. Click Manage categories 3. Select the function you want to use (add, edit, search, delete, view) 4. Go to the page containing the selected function 5. Add/edit/search/delete/view categories		
Alternative flow/process	1a. Cannot login to the system 2a. Show notification 3a. Return to home page		
	1b. Cannot find this categories 2b. Show notification 3b. Return to manage categories		
Error situations	1. Cannot connect to the server 2. Account does not exist 3. Account has not been granted access 4. Password or username entered is incorrect 5. Server crash 6. Categories does not exist		
System state in error situations	- Still can't login - System down		

Table 5: Use case description for manage course functions

Name	Manage course	**Code**	CourseID
Description	Staff manage course: Create, edit, delete, view, search, assign trainee		
Actor	Staff	**Trigger**	Enter manage course
Pre-condition	Staff must login in system		
Post-condition	Switch to manage course page		
Standard flow/process	1. Login in system 2. Click Manage course 3. Select the function you want to use (add, edit, search, delete, view, assign trainee) 4. Go to the page containing the selected function 5. Add/edit/search/delete/view course or assign trainee		
Alternative flow/process	1a. Cannot login to the system 2a. Show notification 3a. Return to home page		
	1b. Cannot find this course 2b. Show notification 3b. Return to manage course		
	1c. Cannot assign this trainee 2c. Show notification 3c. Return to manage course		
Error situations	1. Cannot connect to the server 2. Account does not exist 3. Account has not been granted access 4. Password or username entered is incorrect 5. Server crash 6. Course does not exist 7. Trainee does not exist		
System state in error situations	- Still can't login - System down		

Table 6: Use case description for manage topic functions

Name	Manage Topic	**Code**	TopID
Description	Staff manage topic: Create, edit, delete, view, search, assign trainer		
Actor	Staff	**Trigger**	Enter manage topic
Pre-condition	Staff must login in system		
Post-condition	Switch to manage topic page		
Standard flow/process	1. Login in system 2. Click Manage topic 3. Select the function you want to use (add, edit, search, delete, view, assign trainer) 4. Go to the page containing the selected function 5. Add/edit/search/delete/view topic or assign trainer		
Alternative flow/process	1a. Cannot login to the system 2a. Show notification 3a. Return to home page		
	1b. Cannot find this topic 2b. Show notification 3b. Return to manage topic		
	1c. Cannot assign this trainer 2c. Show notification 3c. Return to manage topic		
Error situations	1. Cannot connect to the server 2. Account does not exist 3. Account has not been granted access 4. Password or username entered is incorrect 5. Server crash 6. Topic does not exist 7. Trainer does not exist		
System state in error situations	- Still can't login - System down		

Table 7: Use case description for manage profile functions

Name	Manage profile	**Code**	#
Description	Staff manage profile: Create, edit, delete, view, search		
Actor	Staff	**Trigger**	Enter manage profile
Pre-condition	Staff must login in system		
Post-condition	Switch to manage profile page		
Standard flow/process	1. Login in system 2. Click Manage profile 3. Select the function you want to use (add, edit, search, delete, view) 4. Go to the page containing the selected function 5. Add/edit/search/delete/view profile		
Alternative flow/process	1a. Cannot login to the system 2a. Show notification 3a. Return to home page		
	1b. Cannot find this profile 2b. Show notification 3b. Return to manage profile		
Error situations	1. Cannot connect to the server 2. Account does not exist 3. Account has not been granted access 4. Password or username entered is incorrect 5. Server crash 6. Profile does not exist		
System state in error situations	- Still can't login - System down		

Table 8: Use case description for Trainer/trainee profile functions

Name	Trainer/trainee profile	**Code**	#
Description	Trainer and trainee can view, search, edit our profile		
Actor	Trainer, trainee	**Trigger**	Enter profile
Pre-condition	Trainer/trainee must login in system		
Post-condition	Switch to profile page		
Standard flow/process	1. Login in system 2. Click profile 3. Select the function you want to use (edit, search, view) 4. Go to the page containing the selected function 5. Edit/search/ view profile		
Alternative flow/process	1a. Cannot login to the system 2a. Show notification 3a. Return to home page		
Error situations	1. Cannot connect to the server 2. Account does not exist 3. Account has not been granted access 4. Password or username entered is incorrect 5. Server crash		
System state in error situations	- Still can't login - System down		

Table 9: Use case description for trainee course functions

Name	Trainee course	**Code**	CourseID
Description	Trainee can view, search course		
Actor	Trainee	**Trigger**	Enter Course
Pre-condition	Trainee must login in system		
Post-condition	Switch to course page		
Standard flow/process	1. Login in system 2. Click course 3. Select the function you want to use (search, view) 4. Go to the page containing the selected function 5. Search/ view course		
Alternative flow/process	1a. Cannot login to the system 2a. Show notification 3a. Return to home page		
	1b. Cannot find this course 2b. Show notification 3b. Return to course page		
Error situations	1. Cannot connect to the server 2. Account does not exist 3. Account has not been granted access 4. Password or username entered is incorrect 5. Server crash 6. Course does not exist		
System state in error situations	- Still can't login - System down		

Table 10: Use case description for trainer topic functions

Name	Trainer topic	**Code**	TopID
Description	Trainer can view, search topic		
Actor	Trainer	**Trigger**	Enter Topic
Pre-condition	Trainee must login in system		
Post-condition	Switch to course page		
Standard flow/process	1. Login in system 2. Click topic 3. Select the function you want to use (search, view) 4. Go to the page containing the selected function 5. Search/ view topic		
Alternative flow/process	1a. Cannot login to the system 2a. Show notification 3a. Return to home page		
	1b. Cannot find this topic 2b. Show notification 3b. Return to topic page		
Error situations	1. Cannot connect to the server 2. Account does not exist 3. Account has not been granted access 4. Password or username entered is incorrect 5. Server crash 6. Topic does not exist		
System state in error situations	- Still can't login - System down		

LO2 Use design and development methodologies with tools and techniques associated with the creation of a business application

P3 Research the use of software development tools and techniques and identify any that have been selected for the development of this application

1. **Research/identify the use of software development tools and techniques**

- **Model - View - Controller (MVC)**

  MVC is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components is built to handle specific development aspects of an application. It is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects. **[8]**

  - **Model** is the component that corresponds to all the data-related logic that the user works. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data.
    - *For example, a Customer object will retrieve the customer information from the database, manipulate it and update its data back to the database or use it to render data.*
  - **View** component is used for all the UI logic of the application.
    - *For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.*
  - **Controllers** act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.
    - *For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.*

The advantage of MVC is the separation of concern. Separation of concern means we divide the application Model, Control and View. Besides, we can easily maintain our application because of the separation of concern. At the same time, we can split many developer's works at a time. It will not affect one developer's work. On the other hand, it also supports test-driven development so we can create an application with unit tests and write our own test cases. The latest version of MVC supports default responsive websites and mobile templates. Therefore, we can create our own view engine, its syntax is very easy to compare to traditional view engines.

The disadvantage of MVC is that we cannot see design page previews like the .aspx page. Every time, we have to run to see its design. In addition, understanding the flow of applications is a very hard one. It is a little bit complex to implement and not suitable for small-level applications. So, its deployment is a little bit hard.

- **Software**

We develop web applications by Visual Studio and use Windows servers for our computers and devices. When working on a Windows server, we can easily set up a backups restoring strategy in Windows and Windows backup is also easy. We also use the Microsoft .NET framework for developing Web applications because it forms with simple and easy. Besides, the caching system in .NET is powerful and easy to use, it is well designed to be able to expand. On the other hand, .NET Core is a cross-platform .NET implementation, it's completely open-source ensures that a broad technical community can continuously contribute to its development. In addition, ASP.NET does an automatic monitoring feature that is built-in. From that, Windows Web Server will closely monitor websites and applications running on it. In the event of any problem like a memory leak or infinite loop, it will immediately alert. This allows us to directly correct these behaviors and create new processes. Mean, monitoring ensures more stability and transparency of .NET applications. Finally, ASP.NET being an MVC-based framework that has a robust infrastructure for authentication, package and routing are integrated with many non-Microsoft technologies such as Bootstrap and AngularJS. So, when using ASP.NET then our website looks greater and makes the web easier display service.

- **Database Server/Cloud Provider**

  LinQ to SQL offers an infrastructure (run-time) for the management of relational data as objects. It is a component of version 3.5 of the .NET Framework and ably does the translation of language-integrated queries of the object model into SQL. These queries are then sent to the database for the purpose of execution. After obtaining the results from the database, LINQ to SQL again translates them to objects.

  Azure SQL Database is an intelligent, scalable, relational database service built for the cloud. Optimize performance and durability with automated, AI-powered features that are always up to date. With serverless compute and Hyperscale storage options that automatically scale resources on-demand, you're free to focus on building new applications without worrying about storage size or resource management.

- **Programming languages**

  Java, PHP, C#, C / C++, SQL, ASP .NET, Objective-C, Visual Basic,...

- **Other**

  We use the entity-relationship diagram to describe the application database and use a unified modeling language class diagram for the core business processing.

2. ERD describing application database

An entity-relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define their properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases. ER diagrams are used to sketch out the design of a database. **[4]**

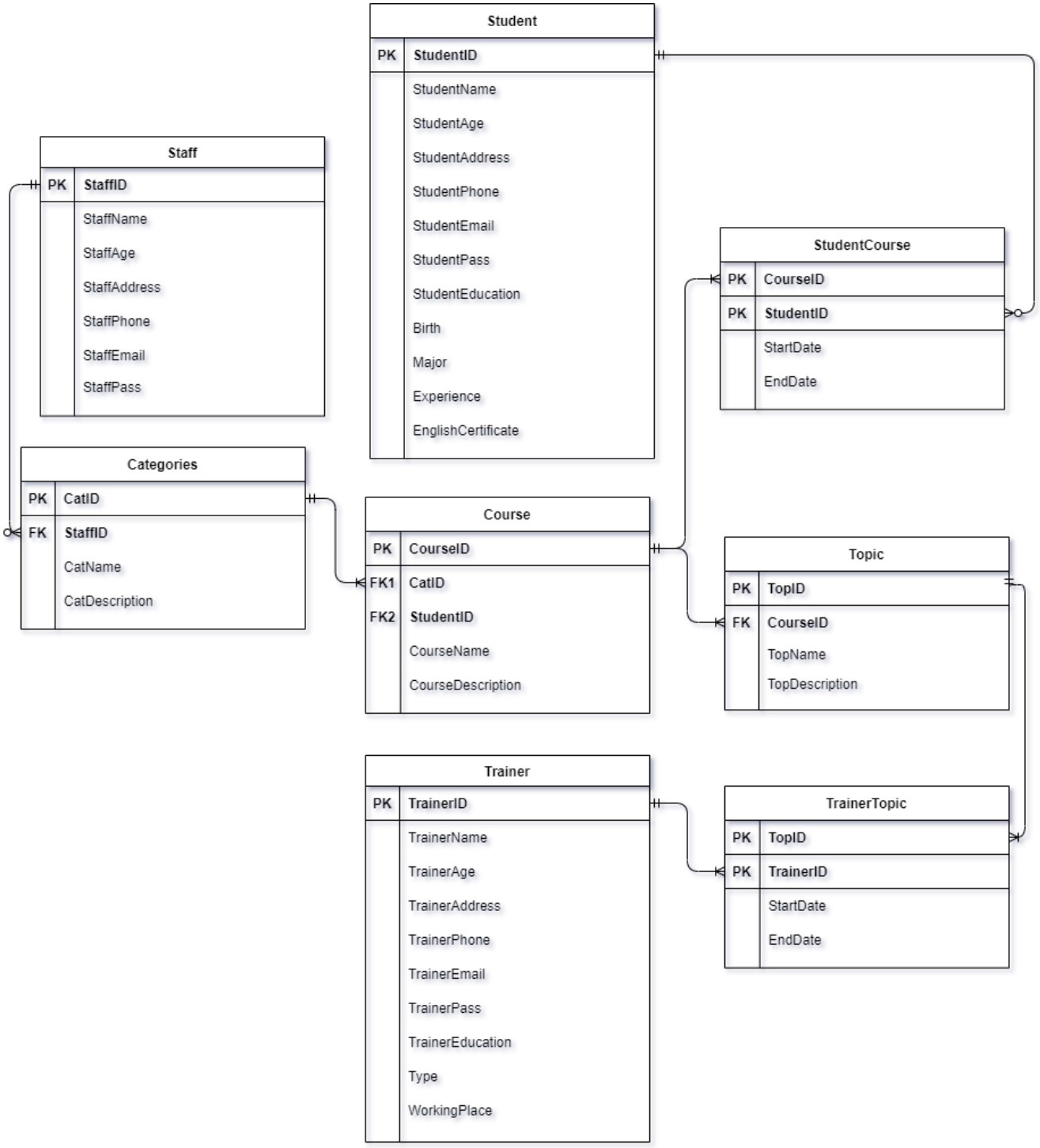

Figure 23: ERD

- Entity description:

  - ✓ Staff (training staff): StaffID, StaffName, StaffAge, StaffAddress, StaffMail, StaffPhone, StaffPass.
  - ✓ Student (trainee): StudentID, StudentName, StudentAge, StudentAddress, StudentMail, StudentPhone, StudentPass, StudentEducation (education level), Birth (date of birth), Major, Experience, EnglishCertificate.
  - ✓ Trainer: TrainerID, TrainerName, TrainerAge, TrainerPass, TrainerAddress, TrainerPhone, TrainerMail, TrainerEducation (education level), Type (external type or internal type), WorkingPlace.
  - ✓ Categories (course categories): CatID (ID of categories), StaffID, CatName (Name of categories), CatDescription (Description of categories).
  - ✓ Course: CourseID, CatID (ID of categories), StudentID (ID of trainee), CourseName, CourseDescription.
  - ✓ Topic: TopID (ID of topic), CourseID, TopName (name of topic), TopDescription (description of topic).

3. UML Class Diagram for the core business processing

The UML Class diagram is a graphical notation used to construct and visualize object-oriented systems. A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. **[5]**

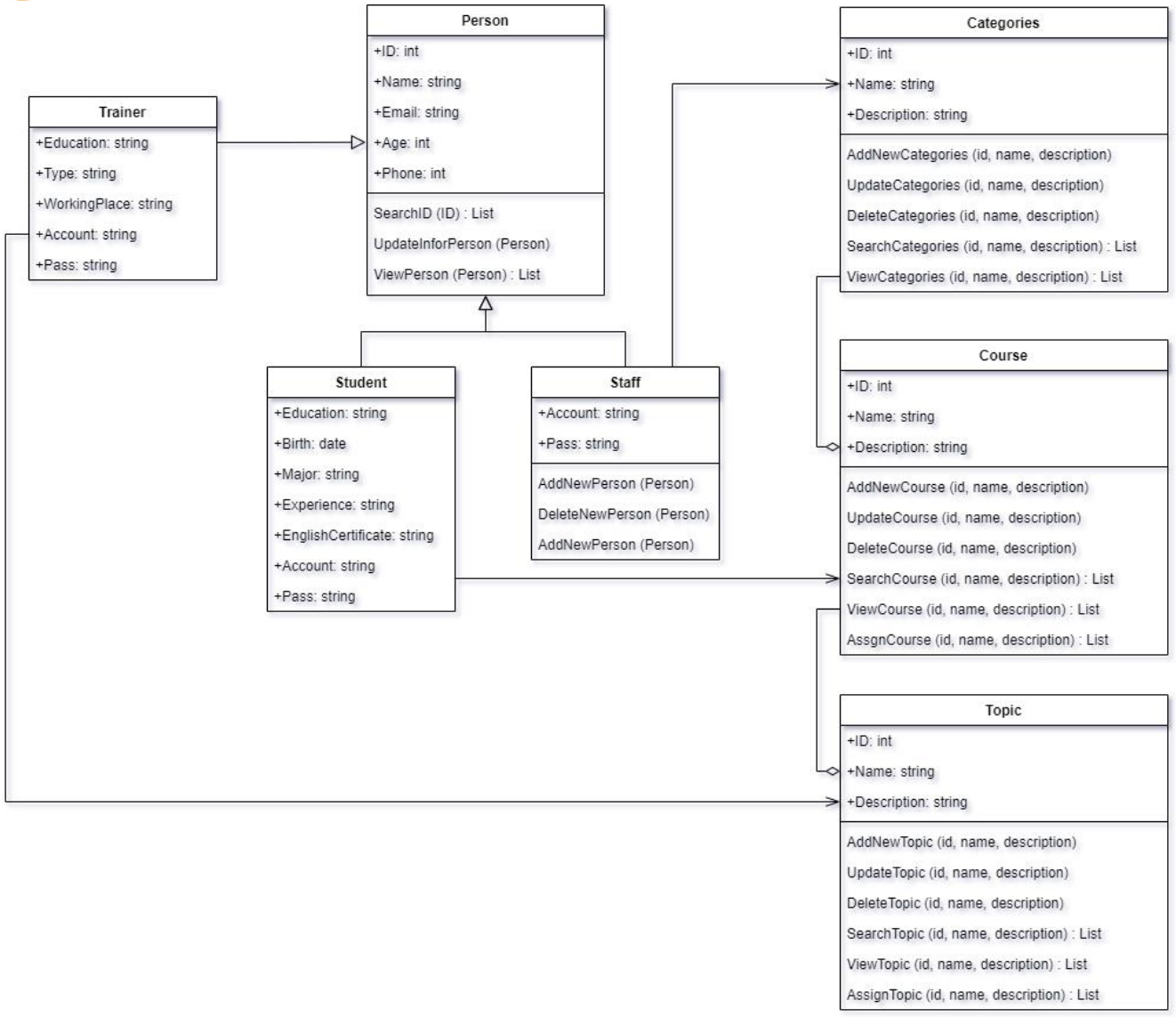

Figure 24: UML Class diagram

M2 Compare the differences between the various software development tools and techniques researched and justify your preferred selection as well as your preferred software development methodology

Software development life cycle (SDLC) is a series of phases that provide a common understanding of the software building process. How the software will be realized and developed from the business understanding and requirements elicitation phase to convert these business ideas and requirements into functions and features until its usage and operation to achieve the business needs. A good software engineer should have enough knowledge on how to choose the SDLC

model based on the project context and the business requirements. **[13]**

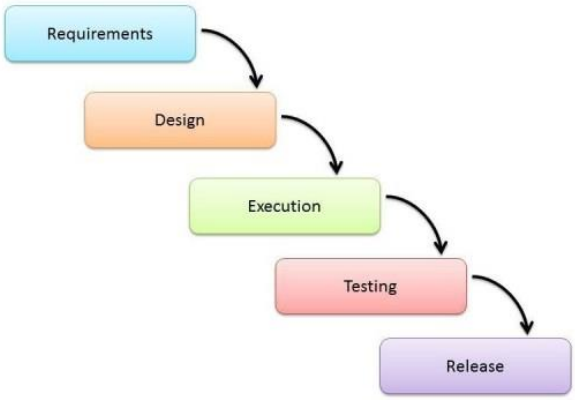

Figure 25: Waterfall Model

The Waterfall Model is a linear sequential flow. In which progress is seen as flowing steadily downwards (like a waterfall) through the phases of software implementation. This means that any phase in the development process begins only if the previous phase is complete. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirements. The waterfall approach is the earliest approach and most widely known that was used for software development. It usage in projects which not focus on changing the requirements, for example, projects initiated from a request for proposals (RFPs), the customer has very clear documented requirements. **[13]**

Table 11: Advantages and Disadvantages of the Waterfall Model

Advantages	Disadvantages
- Easy to explain to the users - Structures approach - Stages and activities are well defined - Helps to plan and schedule the project - Verification at each stage ensures early detection of errors/misunderstandings - Each phase has specific deliverables	- Assumes that the requirements of a system can be frozen - Very difficult to go back to any stage after it finished - A little flexibility and adjusting scope is difficult and expensive - Costly and required more time, in addition to the detailed plan

Prototyping Model refers to the activity of creating prototypes of software applications, for example, incomplete versions of the software program being developed. It is an activity that can occur in software development and it is used to visualize some component of the software to limit the gap of misunderstanding the customer requirements by the development team. This also will reduce the iterations that may occur in the waterfall approach and hard to be implemented due to the inflexibility of the waterfall approach. So, when the final prototype is developed, the requirement is considered to be frozen. This process can be used with any software developing a life cycle model. While this shall be chosen when you are developing a system that has user interactions. So, if the system does not have user interactions, such as a system does some calculations shall not have prototypes. **[13]**

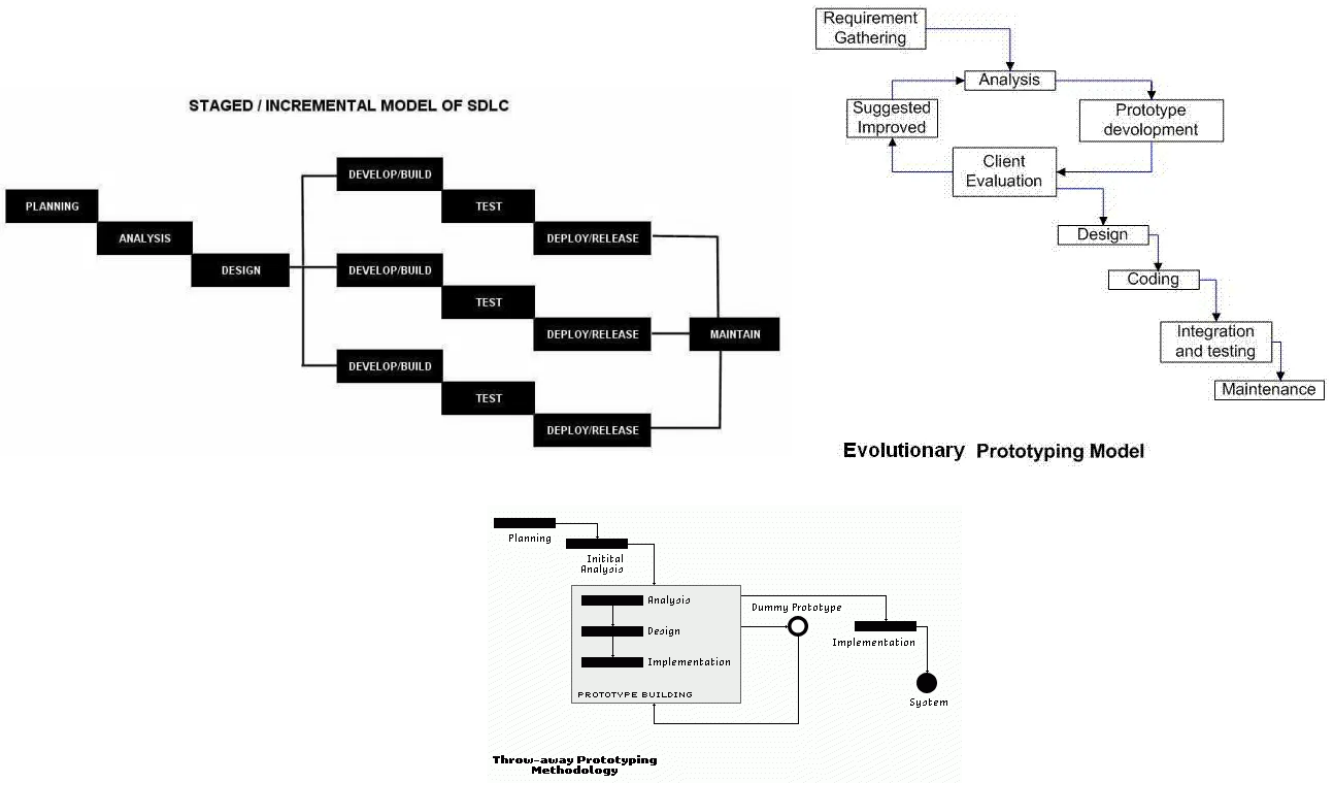

Figure 26: Prototyping Model

Table 12: Advantages and Disadvantages of Prototyping Model

Advantages	Disadvantages
- Reduced time and costs, but this can be a disadvantage if the developer loses time in developing the prototypes - Improved and increased user involvement	- Insufficient analysis. User confusion of prototype and finished system. - Developer misunderstanding of user objectives - Excessive development time of the prototype - It is costly to implement the prototypes

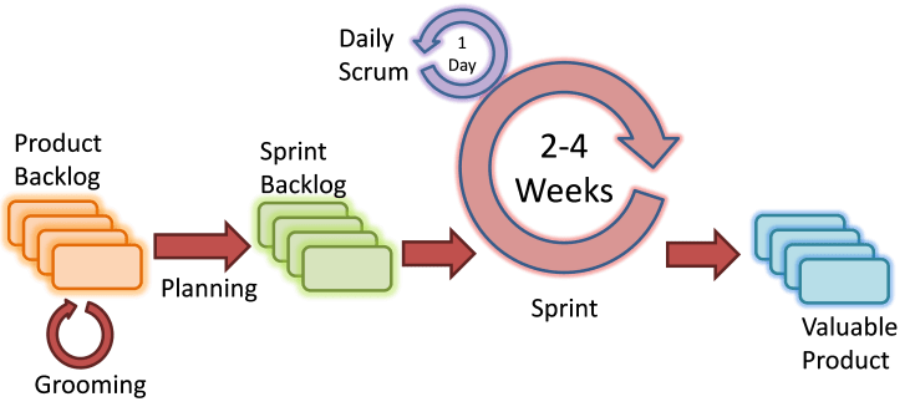

Figure 27: Agile Model

Agile Model based on iterative and incremental development, where requirements and solutions evolve through collaboration between cross-functional teams. It can be used with any type of project, but it needs more engagement from the customer and to be interactive. Also, we can use it when the customer needs to have some functional requirement ready in less than three weeks and the requirements are not clear enough. This will enable more valuable and workable pieces for software early which also increases customer satisfaction.

Table 13: Advantages and Disadvantages of Agile Model

Advantages	Disadvantages
- Decrease the time required to avail of some system features - Face to face communication and continuous inputs from customer representative leaves no space for guesswork - The end result is the high-quality software in the least possible time duration and satisfied customer	- Scalability - The ability and collaboration of the customer to express user needs - Documentation is done at later stages - Reduce the usability of components - Needs special skills for the team

LO1 & 2

D1 Justify the tools and techniques chosen to realize a custom-built website. Justify your preferred selection of tools and techniques in deducing an appropriate solution to a business-related problem

We have come to the conclusion that with the quicker development, testing, and constant feedback from the user, the Agile methodology becomes the appropriate approach for the projects to be delivered in a short span of time. Agile methods also emphasize working software as the primary measure of progress. Combined with the preference for face-to-face communication, agile methods produce very little written documentation relative to other methods.

Performance testing of a web application is important to ensure a seamless user experience when loading pages and data. Some aspects of web performance that we should consider during development. Page weight is the total number of page contents that are HTML, CSS, JavaScript, images, etc. It can usually be measured in kilobytes, but it can also be measured in megabytes, and the useful property of the page load metric is that it is independent of servers and infrastructure, which we can easily measure. As the project progresses. Therefore, we can measure the weight of each packet that the server sends to the client. Perhaps we should reduce the amount of data transferred so that we can speed up the page load time without waiting for users. Another web

performance issue is a disabled cache policy. Our caching strategy must be optimized to achieve efficient data processing by setting the maximum step end heading in the server. We guarantee that during repeated hits, users can reuse the resources they previously downloaded. Ideally, we should aim to secure as many resources as possible and provide authentication tokens. In addition, we can check code compatibility so that we know that any unused scripts or files are being sent to the wrong user and can reduce web performance. So we will optimize the data sent from the server that is used, and none of that data is redundant. There are many security issues that can be addressed with our application, which can lead to system crashes and other negative consequences. We need to apply some of the most common security attacks like SQL injection, XSS, CSFR, etc. to keep the application safe for users at all times.

Finally, we use UML diagrams because it most used and it very flexible. Besides, if using it then the software architecture must be communicated effectively and we need to know only a fraction of the language to use it. In addition, tools in UML very abundant.

REFERENCES

[1] Guru99.com. 2021. *UML Use Case Diagram: Tutorial with EXAMPLE*. [online] Available at: <https://www.guru99.com/use-case-diagrams-example.html> [Accessed 18 April 2021].

[2] Agilemodeling.com. 2021. *User Interface Flow Diagrams (UI Storyboards): An Agile Introduction*. [online] Available at: <http://agilemodeling.com/artifacts/uiFlowDiagram.htm> [Accessed 18 April 2021].

[3] Experience UX. 2021. *What is wireframing | Experience UX*. [online] Available at: <https://www.experienceux.co.uk/faqs/what-is-wireframing/> [Accessed 18 April 2021].

[4] Smartdraw.com. 2021. *Entity Relationship Diagram (ERD) - What is an ER Diagram?*. [online] Available at: <https://www.smartdraw.com/entity-relationship-diagram/> [Accessed 18 April 2021].

[5] Visual-paradigm.com. 2021. *UML Class Diagram Tutorial*. [online] Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/> [Accessed 18 April 2021].

[6] Mindtools.com. 2021. *Risk Analysis and Risk Management: Evaluating and Managing Risks*. [online] Available at: <https://www.mindtools.com/pages/article/newTMC_07.htm> [Accessed 18 April 2021].

[7] Mike Griffiths, 2021. *The Top Five Software Project Risks*. [online] Project Smart. Available at: <https://www.projectsmart.co.uk/top-five-software-project-risks.php> [Accessed 18 April 2021].

[8] Tutorialspoint.com. 2021. *MVC Framework - Introduction - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm> [Accessed 18 April 2021].

[9] Vignesh Mani, 2021. *Advantage and Disadvantage of ASP.NET MVC*. [online] C-sharpcorner.com. Available at: <https://www.c-sharpcorner.com/blogs/advantage-and-disadvantage-of-asp-net-mvc1> [Accessed 18 April 2021].

[10] Tutorialspoint.com. 2021. *LINQ - SQL - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/linq/linq_sql.htm> [Accessed 18 April 2021].

[11] Azure.microsoft.com. 2021. *SQL Database – Managed Cloud Database Service | Microsoft Azure*. [online] Available at: <https://azure.microsoft.com/en-au/services/sql-database/> [Accessed 18 April 2021].

[12] ArtSoft Consult. 2021. *Tools and technologies used in software development at ArtSoft Consult*. [online] Available at: <https://www.artsoft-consult.ro/areas-of-application/tools-and-technologies> [Accessed 18 April 2021].

[13] Sami, M., 2021. *Software Development Life Cycle Models and Methodologies - Mohamed Sami*. [online] Mohamed Sami. Available at: <https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/> [Accessed 18 April 2021].

[14] NATALYA RAHMANY, 2021. *Software Testing Methodology V-Model, Agile, Waterfall, Spiral*. [online] Productive Hut. Available at: <https://productivehut.com/v-model-agile-waterfall-spiral/> [Accessed 18 April 2021].

[15] Microsoft. 2021. *ASP.NET | Open-source web framework for .NET*. [online] Available at: <https://dotnet.microsoft.com/apps/aspnet> [Accessed 18 April 2021].