THE UNIVERSITY OF DA NANG
UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

GRADUATION PROJECT THESIS

MAJOR: INFORMATION TECHNOLOGY

SPECIALTY: INFORMATION SYSTEM

PROJECT TITLE:

DUT JOB - AN APPLICATION CONNECTING DUT'S STUDENTS AND ENTERPRISES

Instructor: **Ph.D. NINH KHANH DUY**
Student: NGUYEN THI THANH TUYEN
Student ID: 102140106
Class: 14T2

Da Nang, 06/2019

THE UNIVERSITY OF DA NANG
UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

GRADUATION PROJECT THESIS

MAJOR: INFORMATION TECHNOLOGY

SPECIALTY: INFORMATION SYSTEM

PROJECT TITLE:

DUT JOB - AN APPLICATION CONNECTING DUT'S STUDENTS AND ENTERPRISES

Instructor: **Ph.D. NINH KHANH DUY**
Student: NGUYEN THI THANH TUYEN
Student ID: 102140106
Class: 14T2

Da Nang, 06/2019

INSTRUCTOR'S COMMENTS

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

REVIEWER'S COMMENTS

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ABSTRACT

Topic title: DUT JOB - An application connecting DUT's students and enterprises.
Student: Nguyen Thi Thanh Tuyen
Student ID: 102140106 Class: 14T2

The aim of this project is to create an application to demonstrate the rationale for connecting enterprise with employability and career landing of DUT's students in Da Nang city.

Students can find a job that is suitable for their field easily. Dozens of job vacancy from enterprises are brought to students and they are public on this website. All transparent information is verified by the admin. Students can be assured without fear of vacancies from scam or multi-level companies. When seeing a position that the students want to apply, they just need to submit a form and their CV are sent to the enterprises holding that recruitment.

I used ASP.NET Core 2.1 framework for building API, Angular 7 and Bootstrap framework for developing the website for users.

DA NANG UNIVERSITY
UNIVERSITY OF SICIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

THE SOCIALIST REPUBLIC OF VIETNAM
Independence - Freedom - Happiness

GRADUATION PROJECT REQUIREMENTS

Student Name: Nguyen Thi Thanh Tuyen Student ID: 102140106

Class: 14T2 Faculty: Information Technology Major: Information Technology

1. *Topic title:*

   DUT JOB - An application connecting DUT's students and enterprises
2. *Project topic :* ☐ *has signed intellectual property agreement for final result*
3. *Initial figure and data:*

   Data crawled from two websites: itviec.com, vietnamwork.com

   Content of the explanations and calculations:
   - Searching and applying for a job online
   - Searching information about enterprises
   - Managing enterprise profile, personal profile
   - The management system for admin
4. *Drawings, charts (specify the types and sizes of drawings):*
   - Use case diagram
   - Sequence diagram
   - Activity diagram
   - Class diagram
5. *Instructor(s):* Ph. D Ninh Khanh Duy
6. *Date of assignment: 26/02/2019*
7. *Date of completion: 04/06/2019*

Da Nang, date month year 2019

Head of Division……. **Instructor**

PHIẾU DUYỆT ĐỒ ÁN TỐT NGHIỆP

ACKNOWLEDGMENTS

I am highly indebted to my supervisor, Ph.D. Ninh Khanh Duy, for his continuous support and guidance throughout the tenure of my project.

I thank my families, who gave me the strength and confidence during my time of learning and during the implementation phase of this project.

I am also grateful to the directions of the management board and engineers of Enclave ODC, Ltd who have facilitated my studying about technology, programming, workplace skill, enhance the best soft skills such as teamwork, attitude and especially English.

Although I have tried my best, it is impossible to avoid mistakes, I hope that lecturers could give me comments and recommends to completely finishing my thesis.

Sincerely,

Nguyen Thi Thanh Tuyen

ASSURANCE

I guarantee:

1. The contents of this senior project are performed by myself following the guidance of my supervisor, Ph.D. Ninh Khanh Duy.

2. All references used in this project thesis, are quoted with the author's name, project name, time, and location to publish clearly and faithfully.

3. All invalid copies, educated statute violation or cheating will be borne the full responsibility by myself.

Sincerely,

Nguyen Thi Thanh Tuyen

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

No.	Items	Description
1	API	Application Programming Interface
2	REST	Representational State Transfer
3	UX/UI	User Experience/ User Interface
4	DUT	Da Nang University of Science and Technology
5	CV	Curriculum Vitae
6	PK	Primary Key
7	FK	Foreign Key
8	admin	Administrator

INTRODUCTION

1. Project overview

1.1. Rationale

In the technology-driven society, Da Nang University of Science and Technology (DUT)'s students are using the internet to land a job. With a click, they can find any job they like that matches their skills as well as their expertise.

Although job-seeking websites offer a range of benefits, there are still retails some drawbacks threats. For example, they will offer a job with a very high salary and a lot of good working conditions, but when it comes to the interview, it is not like that anymore. If the jobs are provided from enterprises, which have been controlled by DUT, it will reduce the risks for students. More than that, the DUT has no tool to synthesize jobs and categorize by faculty for students.

For that reason, I decided to do the project "DUT JOB - An application connecting DUT's students and enterprises."

1.2. Purpose

This is a website application for connecting DUT's students and enterprises. With this system, applying for jobs or finding suitable candidates has never so easy and safe.

You are a DUT's student, you are landing a job, this application is the best choice for you. Diversity of jobs is categorized by faculties (Information Technology, Electrical, Environment, etc.), by levels (internship, fresher, etc.), so it is easy for you to find. When searching a position which fit your abilities, you can apply right away instead of writing a job application that is too outdated. Once you apply, your profile will be sent to the enterprise via mail and let prepare your knowledge well because of calls for the interviews are upcoming soon.

Moreover, this website also allows DUT's students to find information about enterprise such as the address, enterprise's overview of enterprise, jobs offered by its, etc.

You are an enterprise, you want employment information is conveyed exact and safe to potential candidates, there is no reason to let you miss DUT JOB. Contacting our

website's admin and an account will be created for you. After that, you can post new hiring posts. Whenever a job-seekers applies for positions your enterprise are hiring, their CV and cover letter will be sent to yours automatically.

2. Structure of the thesis

INTRODUCTION – This chapter gives information about the rationale, and purpose of the project as well as giving the scope of the problems which will be focused on the thesis.

Chapter 1: THEORIES AND TECHNOLOGIES – This chapter introduces all knowledge theories and technologies used in this project.

Chapter 2: ANALYSIS AND DESIGN – This chapter covers the main features, software requirement specifications and database design of the project.

Chapter 3: IMPLEMENTATION AND EVALUATION – This chapter shows an implementation of this project, including pictures and a brief explanation for each main function.

CONCLUSION - The concluding section of the project simultaneously emphasizes the problem solved, as well as presenting issues still unresolved and provides recommendations and suggestions.

REFERENCES – Presentation about the detail of the referenced information used.

Chapter 1 : THEORIES AND TECHNOLOGIES

This system applies the current popular technologies and platforms that combine to create a system responds to the services that user desire.

To learn more about the technologies, I would like to present the concept of some key technologies that I used in this project.

1.1. ASP.NET Core

1.1.1. Introduction

ASP.NET Core is a cross-platform, high-performance, open-source framework for building modern, cloud-based, Internet-connected applications. With ASP.NET Core, you can [1]:

- Build web apps and services, IOT apps, and mobile back-ends.
- Use your favorite development tools on Windows, mac-OS, and Linux.
- Deploy to the cloud or on-premises.
- Run on .NET Core or .NET Framework.

ASP.NET Core provides the following benefits [1]:

- A unified story for building web UI and web APIs.
- Architected for testability.
- Razor Pages makes coding page-focused scenarios easier and more productive.
- Ability to develop and run on Windows, mac-OS, and Linux.
- Open-source and community-focused.
- Integration of modern, client-side frameworks and development workflows.
- A cloud-ready, environment-based configuration system.
- Built-in dependency injection.
- A lightweight, high-performance, and modular HTTP request pipeline.
- Ability to host on IIS, Nginx, Apache, Docker, or self-host in your own process.
- Side-by-side app versioning when targeting .NET Core.
- Tooling that simplifies modern web development.

1.1.2. Architecture

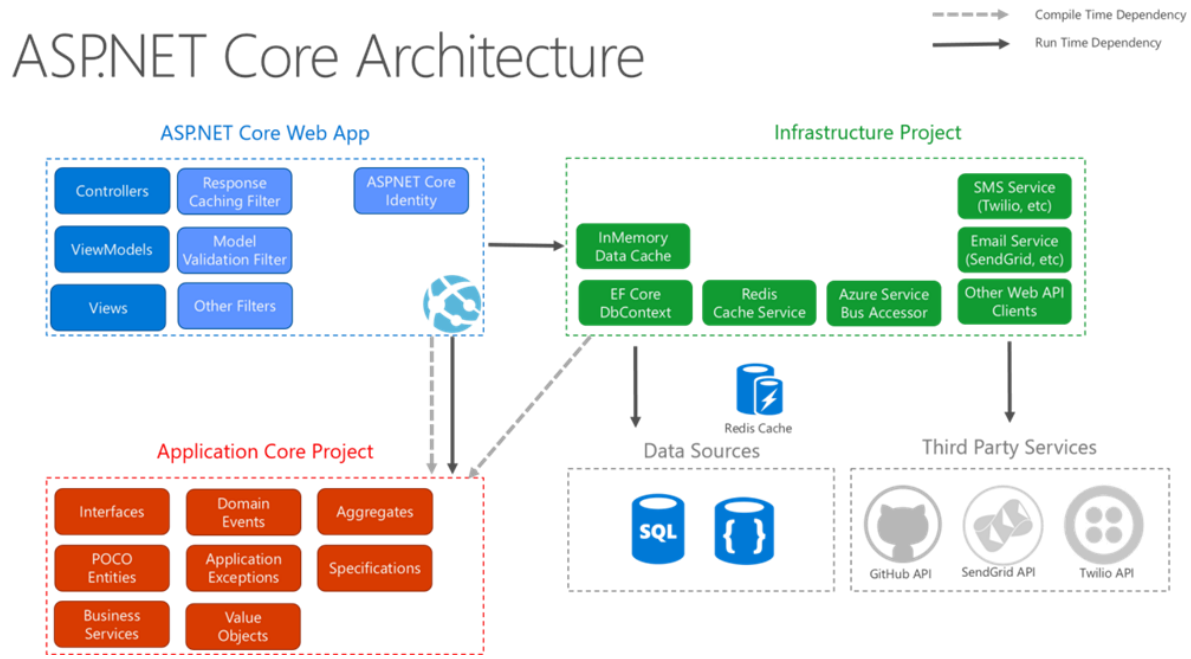

Figure 1.1 ASP.NET Core architecture

The ideology behind ASP.NET Core in general, as the name suggests, is to lay out web logic, infrastructure, and core components from each other in order to provide a more development-friendly environment. The concept is somewhat similar to "N" tier/layer architecture, the only difference is that ASP.NET Core defines the layers as the core component of the platform which relieves the developer from redefining it in order to make a solution more modular and reusable. What happens in ASP.NET Core is that the main business logic and UI logic are encapsulated in ASP.NET Core Web App layer, while the database access layer, cache services, and web API services are encapsulated in infrastructure layer and common utilities, objects, interfaces, and reusable business services are encapsulated as micro-services in application core layer [2].

So, in essence, ASP.NET Core creates necessary pre-defined "N" tier/layers architecture for us developers automatically, which saves our time and effort to worry less about the complexity of necessary "N" tier/architecture of the web project and focus more on the business logic. In non-ASP.NET Core environment, we as developers are more focused on the business logic and the selection of best design patterns, not on the architecture side because, at the end of the day, the final version of our web project is

deployed on a single tier machine. We are more focused on managing the complexity of the project in terms of code re-usability and modular components but on a single tier, very few organizations/enterprises actually implement "N" tier/layer architecture to make their product more robust and modular because it requires more resources consumption, management for scaling the project as necessary, and, of course, hiring of more skilled resources who can build the necessities [2].

So, here comes ASP.NET Core that brings the benefit of a pre-built architectural framework that eases out tier deployment of the project along with providing pre-build Single Page Application (SPA) design pattern, Razor design pattern, and traditional MVC (View-based model) design pattern. These design patterns are mostly used in a hybrid manner but can be utilized as an individual-only pattern as well.

1.2. Angular 7

1.2.1. Introduction

Angular 7 is a platform that makes it easy to build applications with the web. **Angular 7** combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. **Angular 7** empowers developers to build applications that live on the web, mobile, or the desktop [3].

1.2.2. Architecture

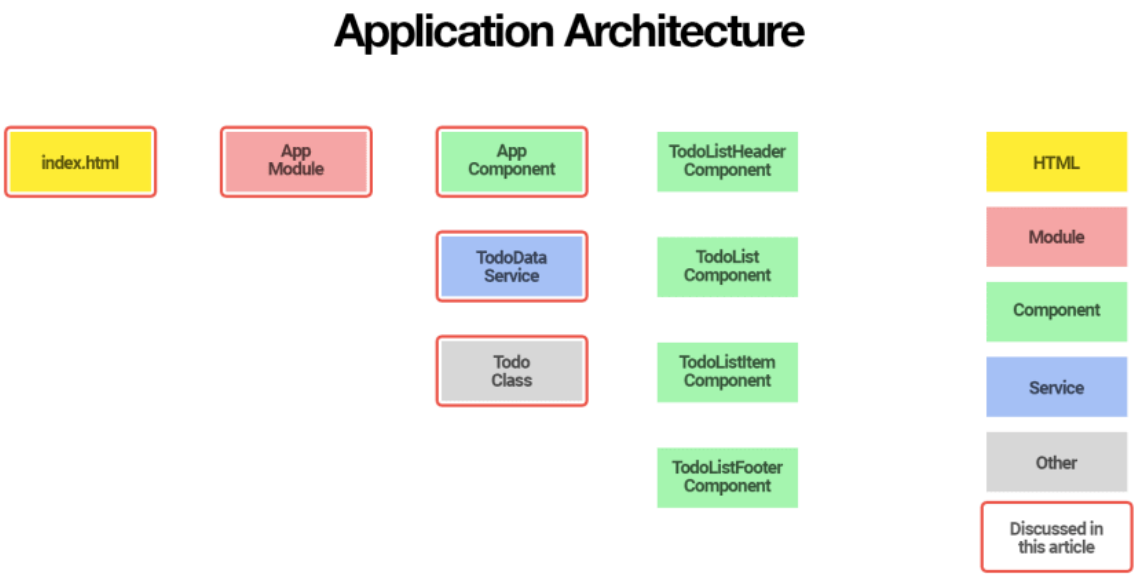

Figure 1.2 Angular architecture

The basic building blocks of an Angular application are *Ng-Modules*, which provide a compilation context for *components*. Ng-Modules collect related code into

functional sets; an Angular app is defined by a set of Ng-Modules. An app always has at least a *root module* that enables bootstrapping and typically has many more feature modules [3].

Components define *views*, which are sets of screen elements that Angular can choose among and modify according to your program logic and data. Components use *services*, which provide specific functionality not directly related to views. Service providers can be *injected* into components as *dependencies*, making your code modular, reusable, and efficient [3].

Both components and services are simply classes, with *decorators* that mark their type and provide metadata that tells Angular how to use them.
The metadata for a component class associates it with a template that defines a view. A template combines ordinary HTML with Angular directives and binding markup that allow Angular to modify the HTML before rendering it for display. The metadata for a service class provides the information Angular needs to make it available to components through *dependency injection (DI)* [3].

1.3. SQL Server

1.3.1. Introduction

SQL Server is a relational database management system, or RDBMS, developed and marketed by Microsoft [4].

SQL Server is built on top of SQL, a standard programming language for interacting with the relational databases. SQL Server 2017 is tied to Transact-SQL, or T-SQL, Microsoft's implementation of SQL that adds a set of proprietary programming constructs.

SQL Server represents a major step towards making SQL Server a platform that gives you choices of development languages, data types, on-premises or cloud, and operating systems by bringing the power of SQL Server to Linux, Linux-based Docker containers, and Windows [4].

1.3.2. Architecture

SQL Server consists of two main components: Database Engine and SQLOS (SQL Operation System) [5]

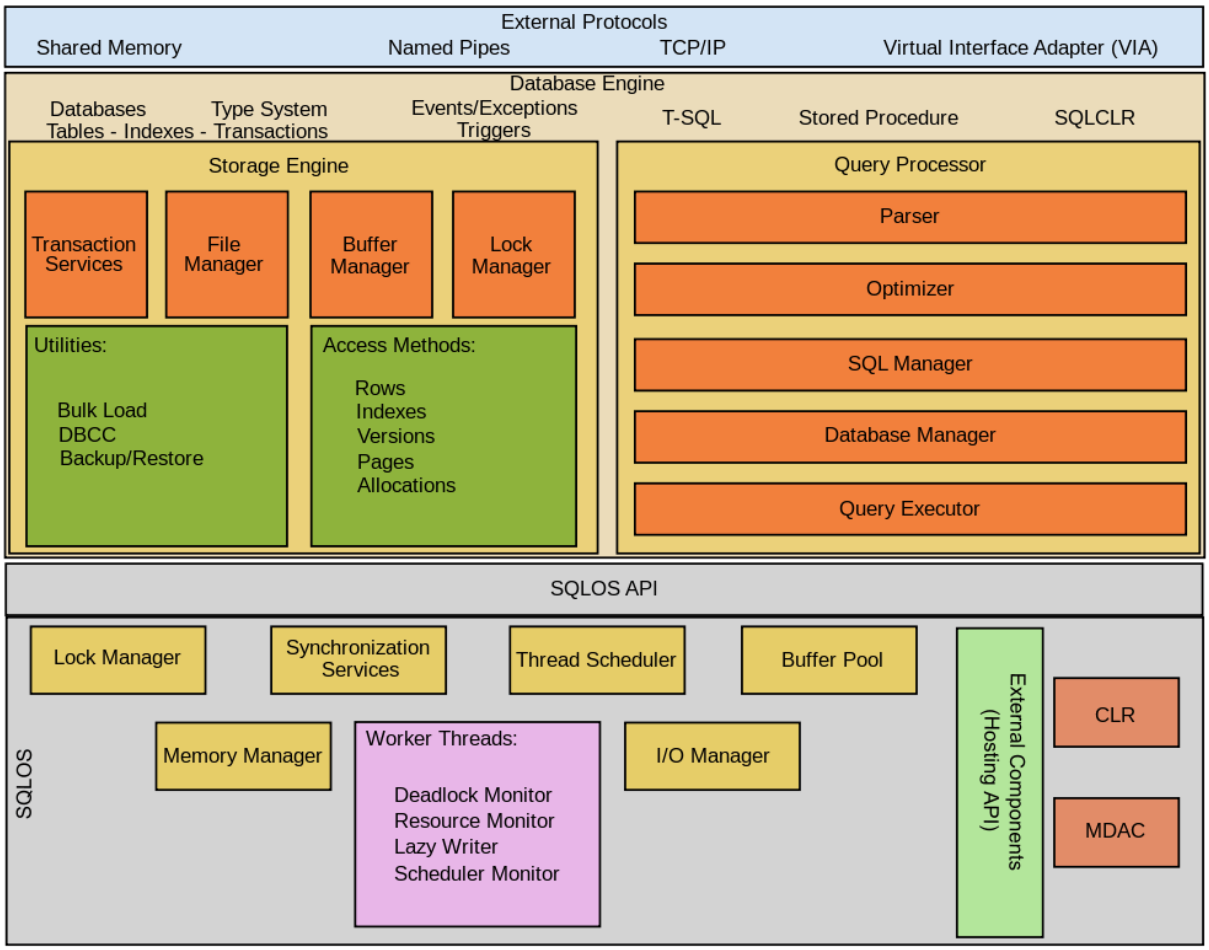

Figure 1.3 SQL Server architecture

❖ **Database Engine:**

- The core component of the SQL Server is the Database Engine. The Database Engine consists of a relational engine that processes queries and a storage engine that manages database files, pages, pages, index, etc. The database objects such as stored procedures, views, and triggers are also created and executed by the Database Engine.
- The Relational Engine contains the components that determine the best way to execute a query. The relational engine is also known as the query processor.
- The relational engine requests data from the storage engine based on the input query and processed the results.
- Some tasks of the relational engine include querying processing, memory management, thread and task management, buffer management, and distributed query processing. The storage engine is in charge of storage and retrieval of data from the storage systems such as disks and SAN.

❖ **SQLOS (SQL Operation System):**

- Under the relational engine and storage engine is the SQL Server Operating System or SQLOS.
- SQLOS provides many operating system services such as memory and I/O management. Other services include exception handling and synchronization services.

1.4. Conclusions

By studying and learning about the above technologies, I successfully applied the concepts and their mechanism operating in this project to create a website.

Some of these technologies are not new, but they're widely using and a trend for the software development industry. Therefore, understanding the concept is very important, help to apply properly technology for each project, in order to improve efficiency and usability.

Chapter 2 : ANALYSIS AND DESIGN

This chapter will go into detail the requirements, describing nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the application. This consists of a package containing Requirements Specification, Use-Cases of the use-case model, Use Case Specifications and Activity Diagram. Shows an overview of what functions the application can satisfy. In addition, it defines the architecture, modules, and data for a system to satisfy specified requirements. System design is intended to be the link between the system architecture and the implementation of technological system elements that compose the physical architecture model of the system. It could be seen as the application of systems theory to product development.

The system design process is to provide sufficiently detailed data and information about the system and it is a system element to enable the implementation consistent with architectural entities as defined in models and views of the system architecture. It shows the components of the application, the structure of data tables, the relationship the elements that make up the system.

2.1. Requirement analysis

2.1.1. *General requirements*

I would like my system to have full of features to support for finding candidates as well as jobs easily.

About enterprises, to login the system, they need to contact admin and an account is created for them to post new hiring information. And they can receive CVs from candidates when they applied via email.

For job seekers, they are more inclined to go to the website and simply see the latest recruitment article. Once they want to apply for somewhere, the thing they need to do is register an account, upload resumes and suitable work will be in their hand.

To let the system work as best as I can, functions such as manage profile (for enterprises and students), manage post (for enterprises), manage skills, technique (for students) are necessary.

And last but not least, the Admin of the page will handle accounts and review all of the hiring posts to ensure safe communication for students and enterprises as well.

2.1.2. Actors

This system has three main actors: guest, student, enterprise, admin.

1) Guest

A guest has functions like:

- Searching enterprises or jobs
- Registering a new account
- Viewing 4 faculties have the most hiring posts: when a guest opens the website application, they will see the top 4 faculties have the most hiring posts, which is ordered based total hiring post.
- Viewing 8 latest hiring posts: when a guest opens the website application, they will see the 8 latest hiring posts which below top 4 faculties.
- Searching and viewing detailed information of enterprises
- Viewing detailed hiring posts

2) Student

A student also has the same functions as a guest. There are also some other functions like:

- Updating personal profile
- Uploading CV
- Adding skills to get an email which has a lot of jobs suitable for you.
- Applying for a job online

3) Enterprise

Enterprise account is created by admin. They have some features like:

- Adding a new hiring post: the enterprise can add their new hiring post, then waiting for administrator check and activated it. If the admin accepts this hiring post, it can display for everyone to see.
- Viewing the hiring posts list
- Updating the enterprise's profile: the enterprise can edit profile information such as enterprise name, phone number, enterprise overview, etc.
- Receiving CV of candidates who apply for jobs which they're managed via email.

4) Admin

Admins have multiple responsibilities that revolve around the maintenance of websites. They are in charge of making sure there are back-ups for all website content, software, and applications in case of a disaster or emergency. They also

make sure that web pages are up to date, secure and function smoothly for the user. In the case of website problems or crashes, the admin is responsible for troubleshooting the problem and fixing it as quickly as possible.

There are the main features of admin: manage student, manage enterprise, manage faculty, etc.

2.2. System analysis

2.2.1. System architecture

This system includes:

- API Service: I use ASP.NET Core 2.1 Framework as the main basic, combining with SQL Server under representational state transfer (REST) technology.
- Customer Website application: I use Angular 7 to build this application. The application is a crucial bridge to connecting between enterprise and student.
- Admin website application: This website assists Administrators to manage the whole system, including statistic.

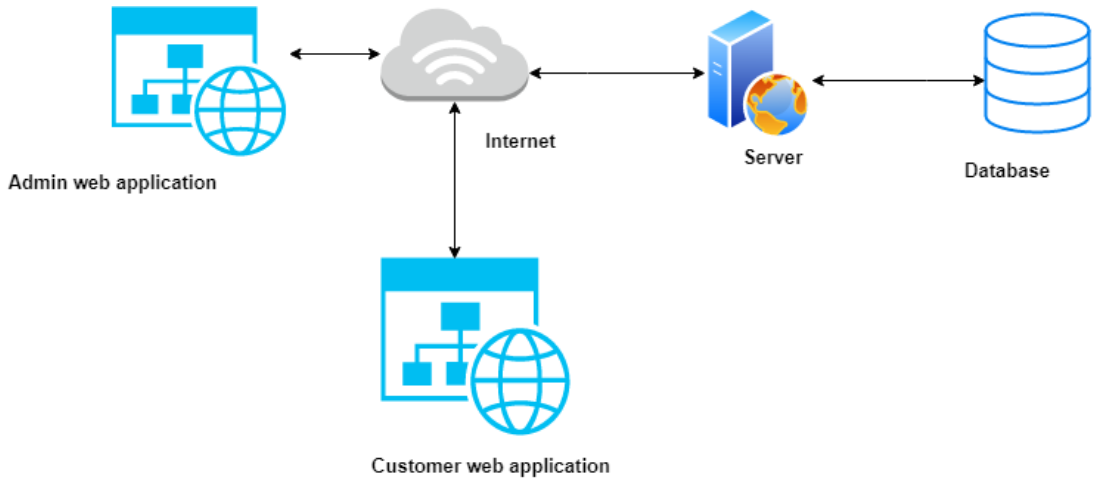

Figure 2.1 System architecture

2.2.2. Use case diagram

1) Guest

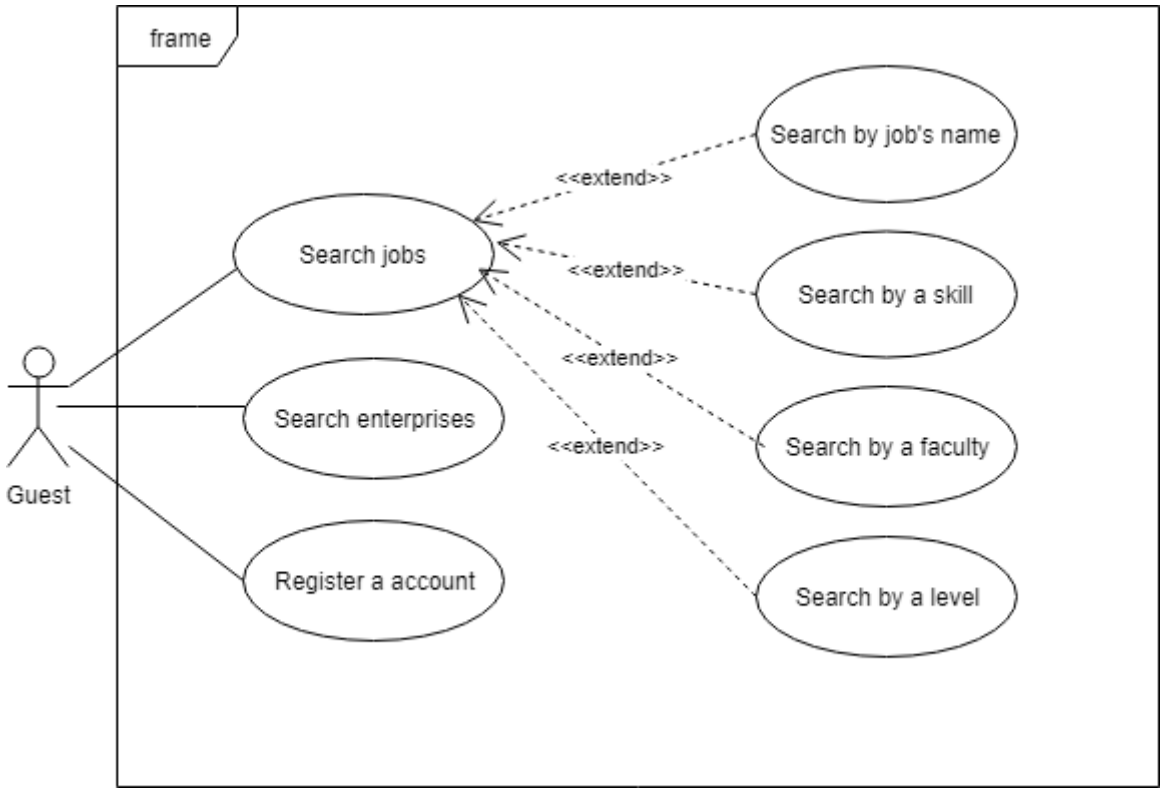

Figure 2.2 Use case of the guest actor

2) Student

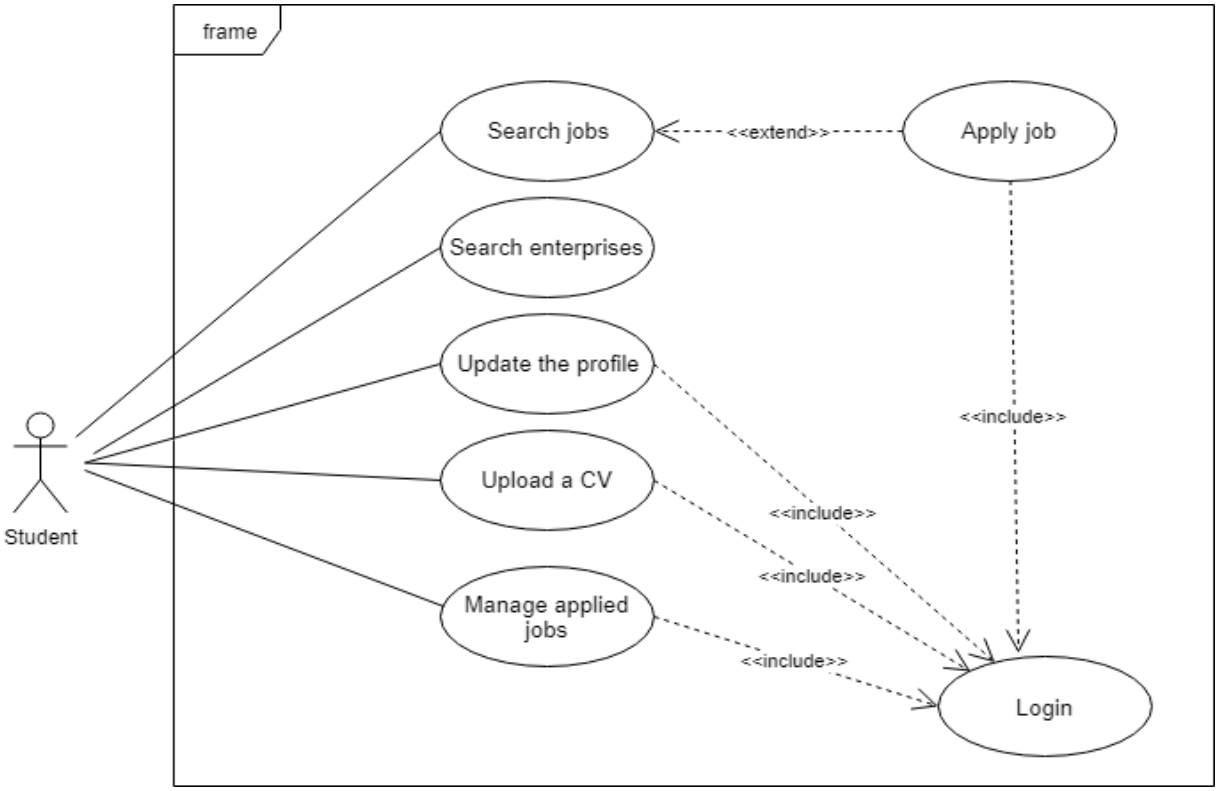

Figure 2.3 Use case of the student actor

3) Enterprise

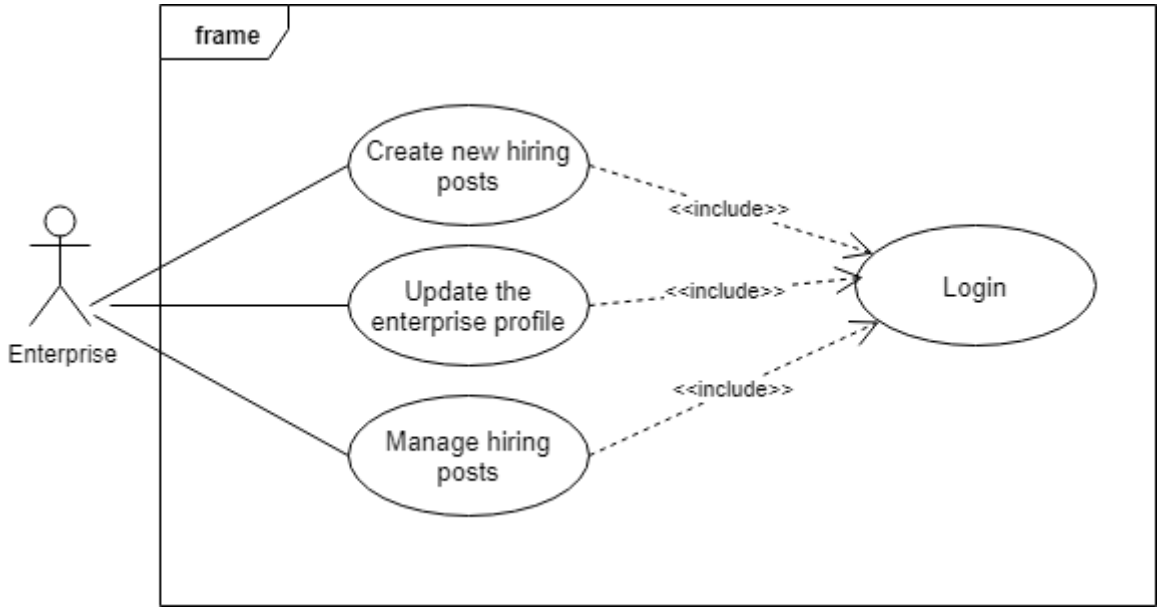

Figure 2.4 Use case of the enterprise actor

4. Admin

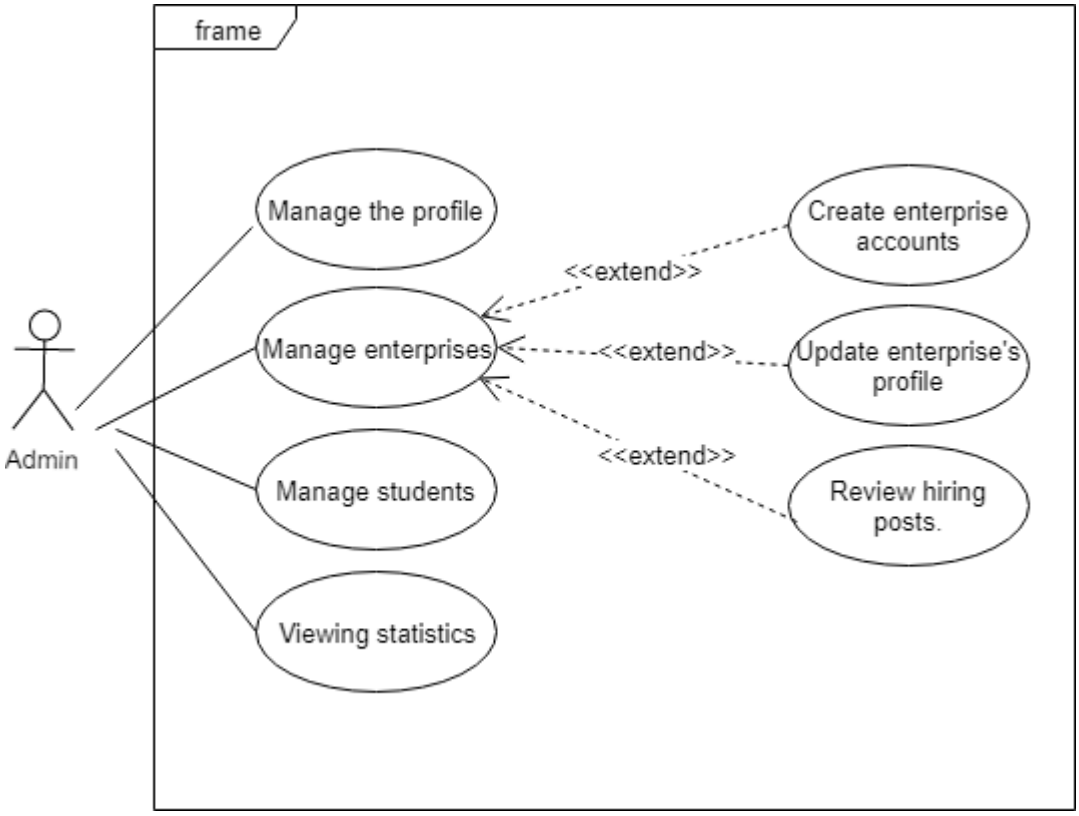

Figure 2.5 Use case of the admin actor

2.2.3. Description for main use cases

1) Register a new account

Table 2.1 Use case description – Register a new account

Use Case ID	UC-01		
Actor	Guest		
Brief description	This use case for register a new account.		
Pre-conditions	None		
Post-conditions	The server received the request from the user then add data into the database. Notify success for user.		
Flow of events		Actor Input	System Response
	1	Click the “Register” button.	
	2		The application shows the register page.
	3	Fill all the information and click the “Register” button.	
	4		- The server received the request from the user, adds data into the database. - Show notification success.

2) Login

Table 2.2 Use case description – Login

Use Case ID	UC-02		
Actor	Student, Enterprise, Admin		
Brief description	This use case for login.		
Pre-conditions	The user who has an account to access the application.		
Post-conditions	User logged in successfully.		
Flow of events		Actor Input	System Response
	1	User opens the website.	
	2	Click the “Login” button at the header.	
	3		The system shows the login form.

	4	User fills the username and password. Click the "Login" button.	
	5		The system validates account that user filled and submit. Show index page (dashboard page with the account of admin) if the user's account is correct else show notification error and return login page.

3) Search jobs

Table 2.3 Use case description – Search jobs

Use Case ID	UC-03		
Actor	All the actor.		
Brief description	This use case for search jobs		
Pre-conditions	The user sends a keyword		
Post-conditions	The server received the request from the user, searching and result response		
Flow of events		Actor Input	System Response
	1	Click the "Find Jobs" button at the index page.	
	2	Enter "keyword" at input search job.	
	3		Job list which fit your keyword.

4) Apply for a job

Table 2.4 Use case description – Apply for a job

Use Case ID	UC-04
Actor	Student
Brief description	This use case for the student to apply for a job.
Pre-conditions	Student already logged into the website.

Post-conditions	The server received the request from the user then add data into the database. Notify to enterprise who manage this job about the new candidate via email.		
Flow of events		Actor Input	System Response
	1	Click the “Apply now” button.	
	2		The application shows the apply job page.
	3	Fill all the information and click the “Apply” button.	
	4		- The server received the request from the student, adds data into the database. - Right away send an email to enterprise who manage this job about the new candidate. - Show notification success.

5) Create a new enterprise

Table 2.5 Use case description – Create a new enterprise

Use Case ID	UC-05		
Actor	Admin		
Brief description	This use case for the admin to add a new enterprise.		
Pre-conditions	Admin already logged into the application.		
Post-conditions	The server received the request from admin, adds data into database.		
Flow of events		Actor Input	System Response
	1	Click the “Add new Company” button at enterprise page.	
	2		Show add new enterprise page.
	3	Fill all the information that is required and click the “Add”	

		button.	
	4		- The server received the request from the admin, adds data into the database. - Right away send an Email to enterprise to notify that they had an account as an enterprise. - Show notification success.

6) Create a new hiring post.

Table 2.6 Use case description – Create a new hiring post

Use Case ID	UC-06		
Actor	Enterprise		
Brief description	This use case for the enterprise to add a new hiring post.		
Pre-conditions	Enterprise already logged into the application.		
Post-conditions	The server received the request from enterprise, adds data into database.		
Flow of events		Actor Input	System Response
	1	Click the “Add new job” button at index page.	
	2		Show add new job page.
	3	Fill all the information that is required and click the “Add” button.	
	4		- The server received the request from the enterprise, adds data into the database. - Show notification success.

7) Update enterprise's profile

Table 2.7 Use case description – Update enterprise's profile

Use Case ID	UC-07		
Actor	Enterprise		
Brief description	This use case for update enterprise's profile.		
Pre-conditions	Enterprise already logged into the application.		
Post-conditions	The server received the request from the enterprise, edits data in the database and update the view page in the website.		
Flow of events		Actor Input	System Response
	1	Click the "Company profile" button at the header of the index page.	
	2		Show the detail information of the company.
	3	Update enterprise's information. Click the "Update" button.	
	4		Update data and show the success notification

8) Update personal information

Table 2.8 Use case description – Update personal information

Use Case ID	UC-08		
Actor	Student		
Brief description	This use case to update personal information.		
Pre-conditions	Student already logged into the application.		
Post-conditions	The server received the request from the student, edits data in the database and update the view page on the website.		
Flow of events		Actor Input	System Response
	1	Click "Your profile" at the header of the index page.	
	2		Show the detail information of the student.

	3	Update personal information Click the “Update” button.	
	4		Update data and show the success notification

2.2.4. Activity diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. Activity diagram is used as a flowchart that consists of activities performed by the system. We must clearly understand the elements used in the activity diagram. The main of an activity diagram is the activity itself. After identifying the activities, we need to understand how they associated with constraints and conditions.

1) Register a new account

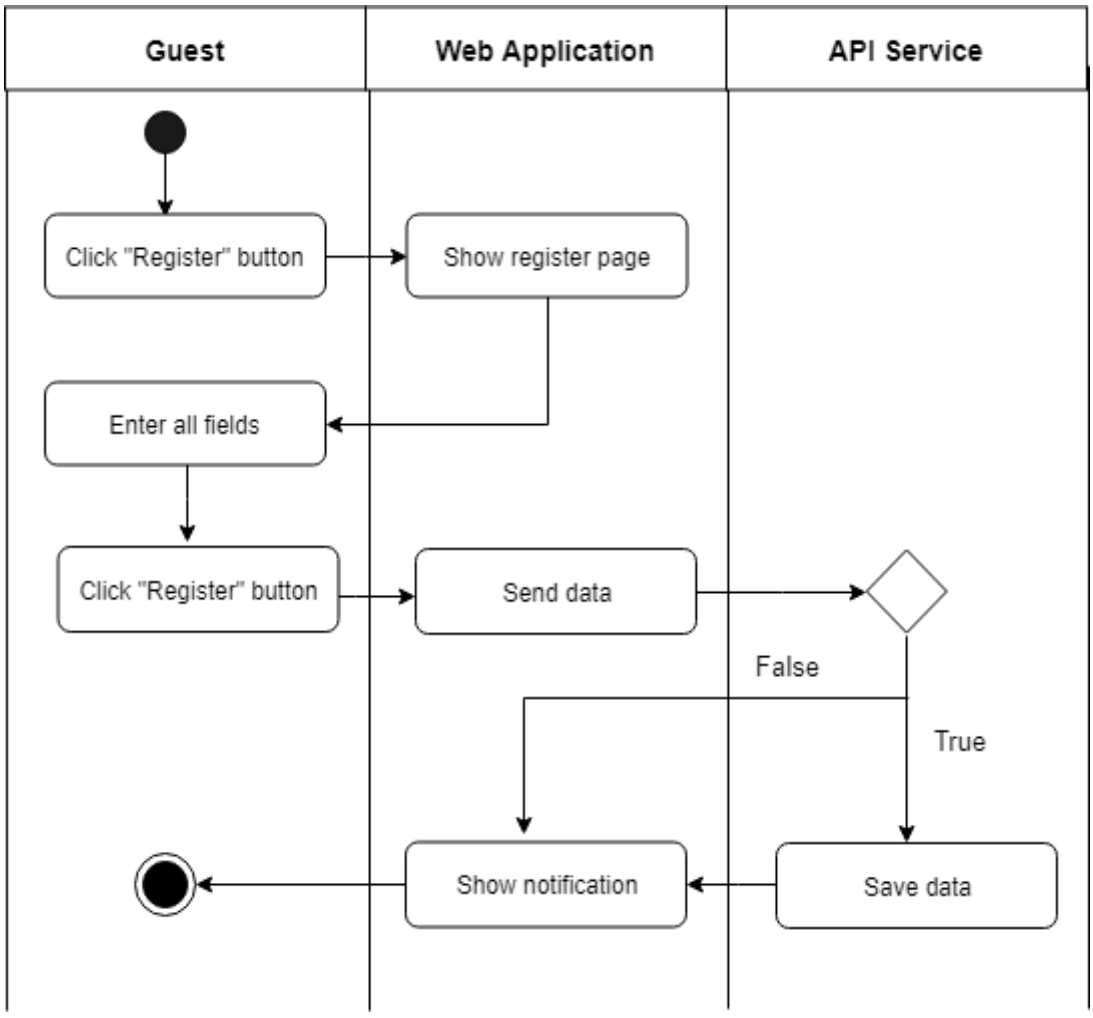

Figure 2.6 Activity diagram – Register a new account

This picture describes the activity of registering a new account as a student account. If you want to have an account to do more (apply for a job, receive an email

for a job that suits your skills, etc.) you only need to register the full form (email and password) then send it. We will check your information if it is ok, we will create an account for you and send you a message that you have succeeded. Otherwise, we give you a message about why your account was not created.

2) Login

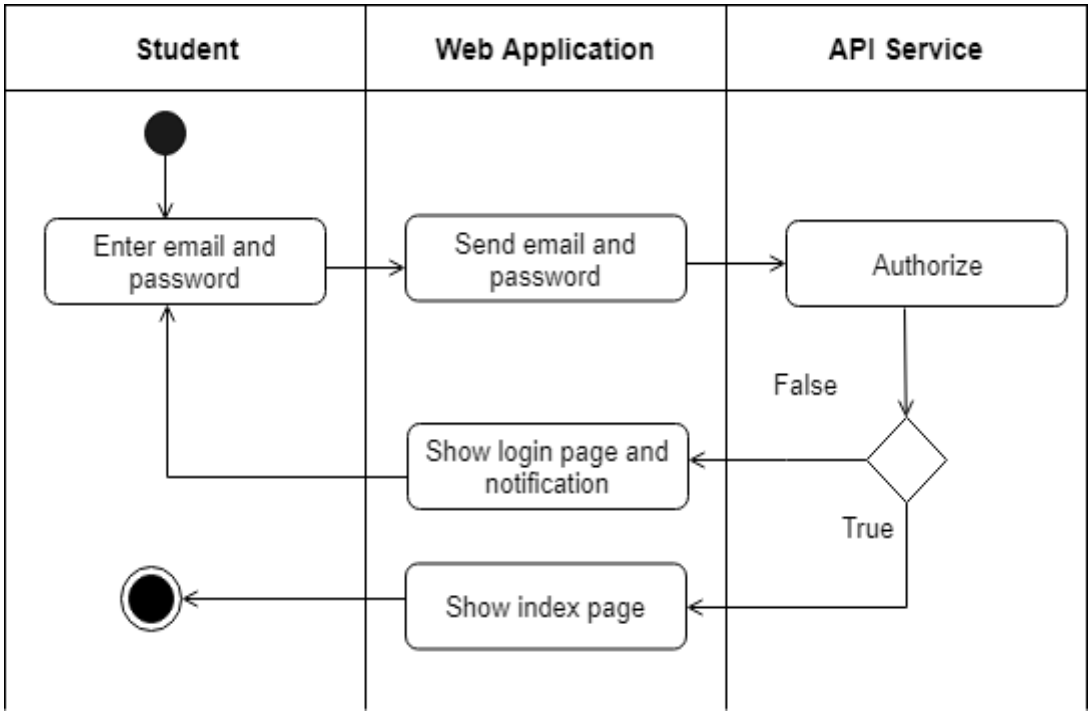

Figure 2.7 Activity diagram – Login

This picture describes the action of logging in. At first, you must enter your email and your password. Then you press the Login button, we will check this information. If it's correct, the system will navigate to the index page.

3) Search jobs

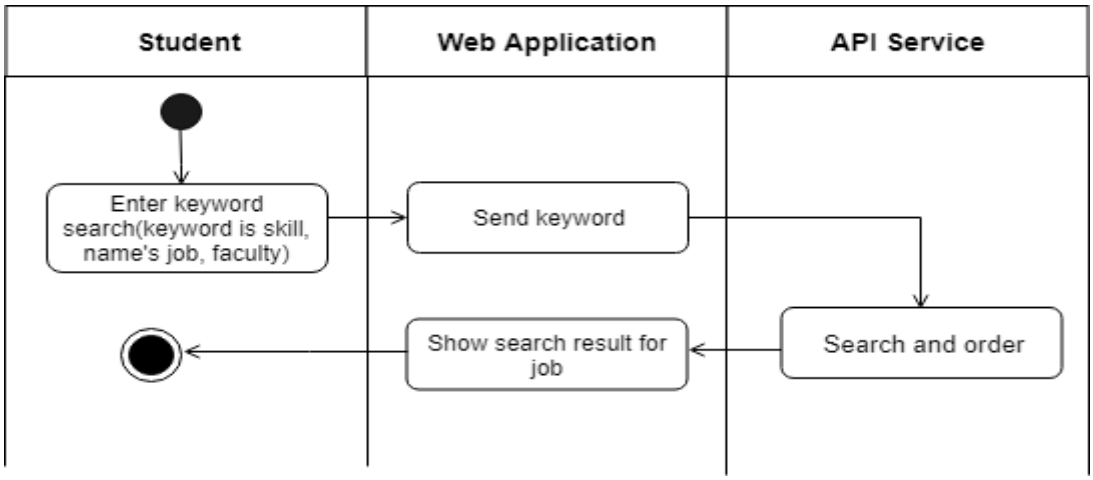

Figure 2.8 Activity diagram – Search jobs

This picture describes the action of finding jobs that match your skills. You only need to enter a keyword such as a skill, a faculty, etc. After that, we will find the jobs that you want. If we can't find any job that matches your keyword, we still give you for some jobs that are closest to the keyword you're looking for.

4) Apply for a job

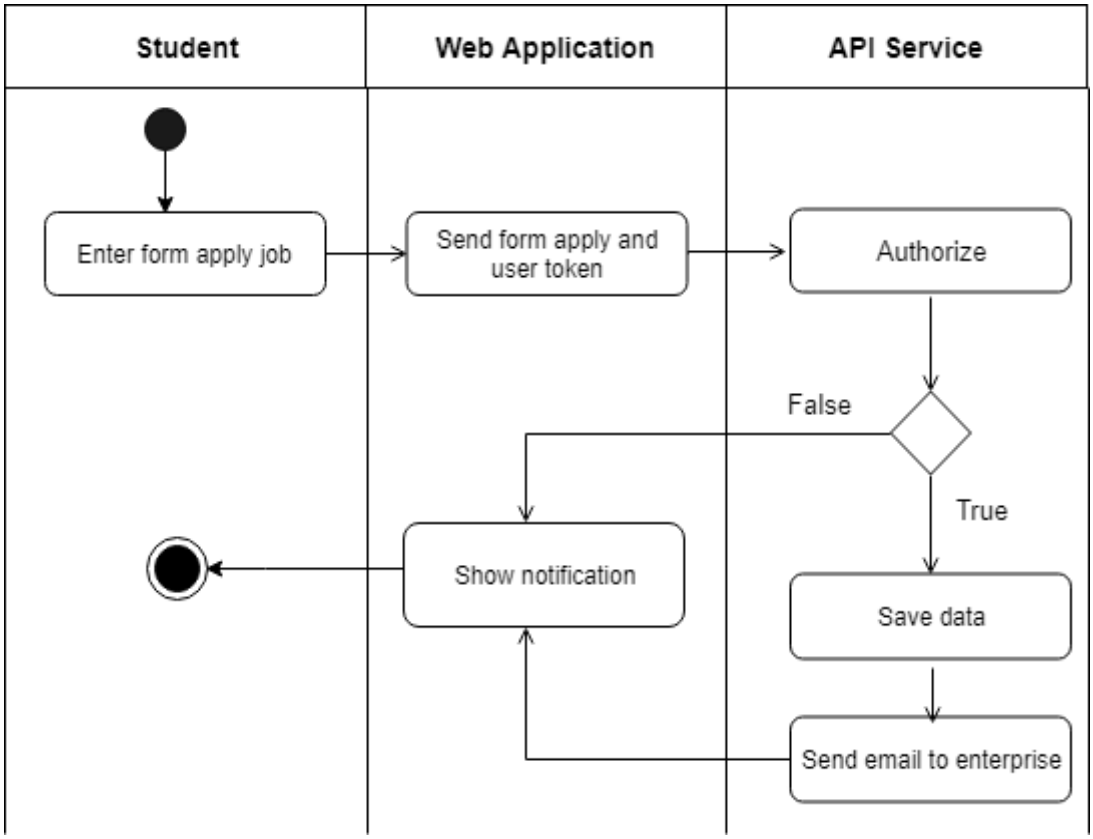

Figure 2.9 Activity diagram – Appy a job

This picture describes the act of applying for a job. You need to fulfill the recruitment form. The form has information as such as your name, your cover letter, and your curriculum vitae. However, you just enter your cover letter. The remaining information we'll get from your profile. After you submit it we will save your form and right away send it to the enterprise which manages this job via email.

The jobs you submitted will be saved. You can easily manage it on this application. You can review the history of what you have submitted and you also can apply for this job again.

5) Create a new enterprise

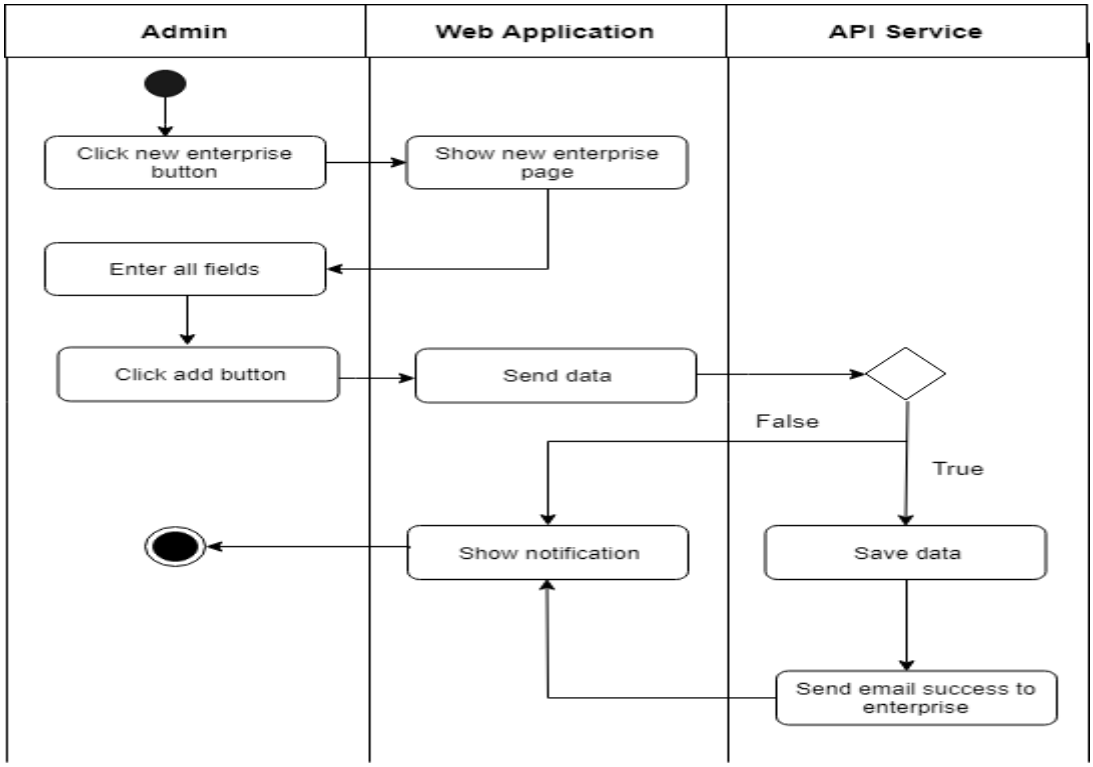

Figure 2.10 Activity diagram – Create a new enterprise

This picture describes the activity of creating a new enterprise. Admin can create a new enterprise through the application. Before creating a new one, the data must be carefully verified.

6) Create a new hiring post

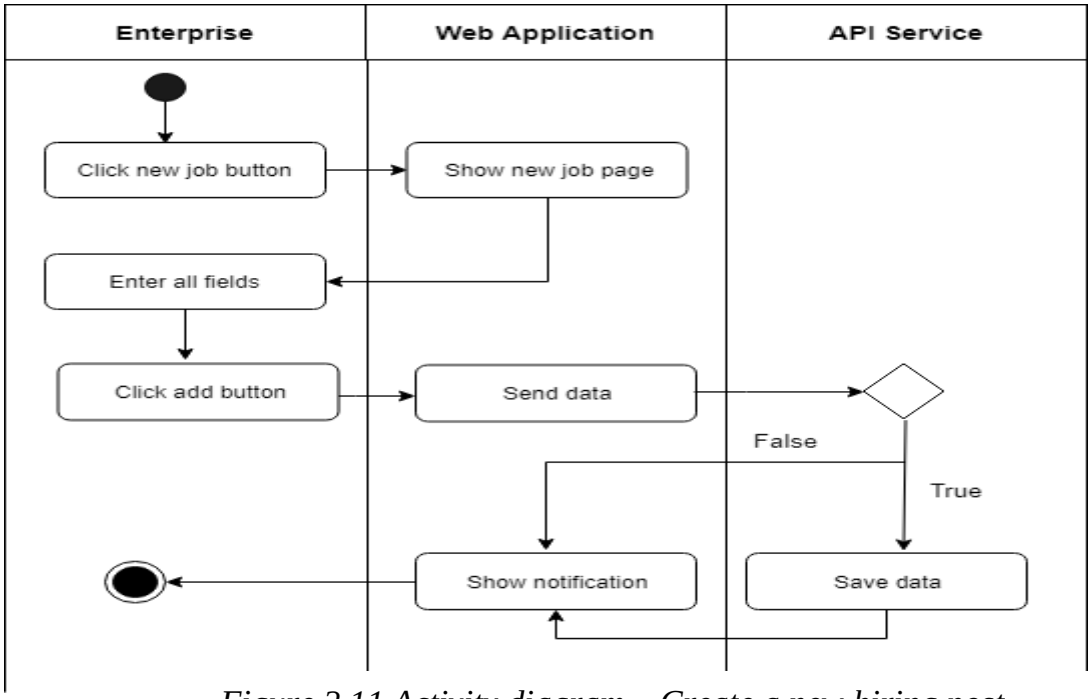

Figure 2.11 Activity diagram – Create a new hiring post

This picture describes the activity of creating a new job. Enterprise can create a new job through the application. Before creating a new one, the data must be carefully verified.

7) View detail an enterprise

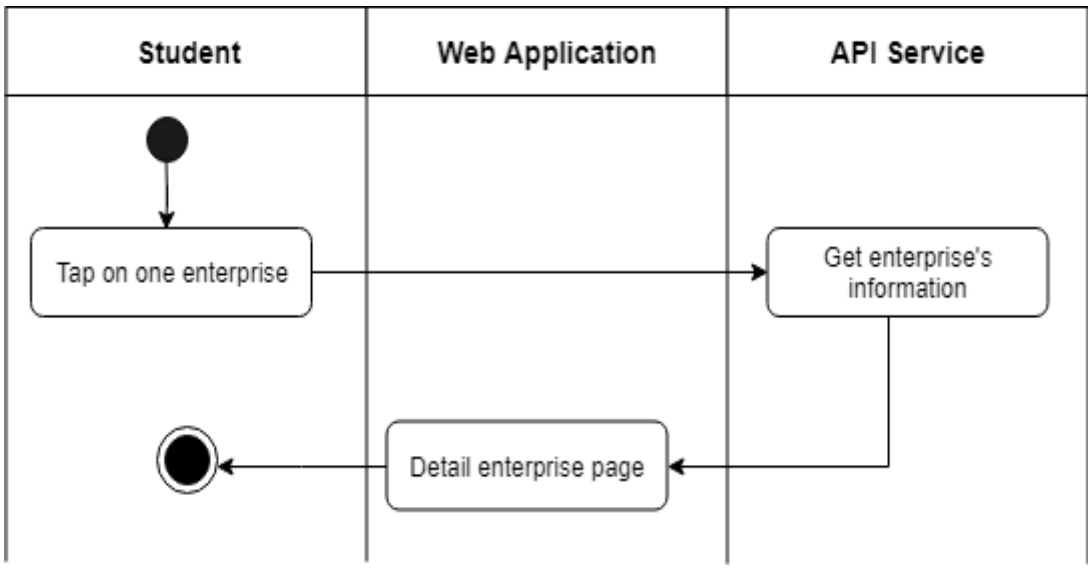

Figure 2.12 Activity diagram – View detail an enterprise

This picture describes the action of show detail for an enterprise. All actor can view an enterprise's information. Just click on enterprise name, all the descriptions of it will be displayed. Details are address, phone, enterprise's overview, jobs offered by its.

8) View detail a job

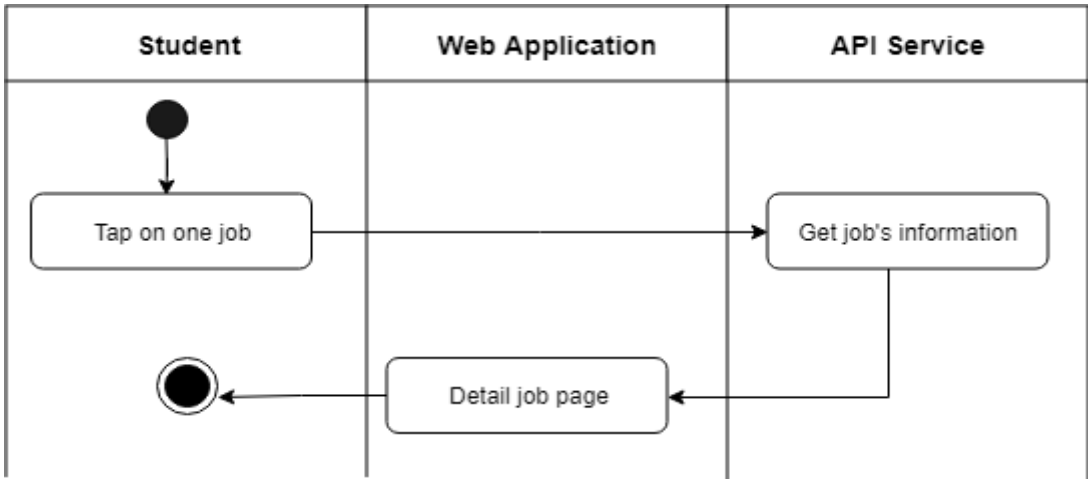

Figure 2.13 Activity diagram – View detail a job

This picture describes the action of show detail for a job. All actor can view a job's information. Just click on the title of the job, all the descriptions of it will be displayed. Details are salary, job description, benefits, details of the company that manages this job, your skills and experience need for it.

9) Update enterprise's profile

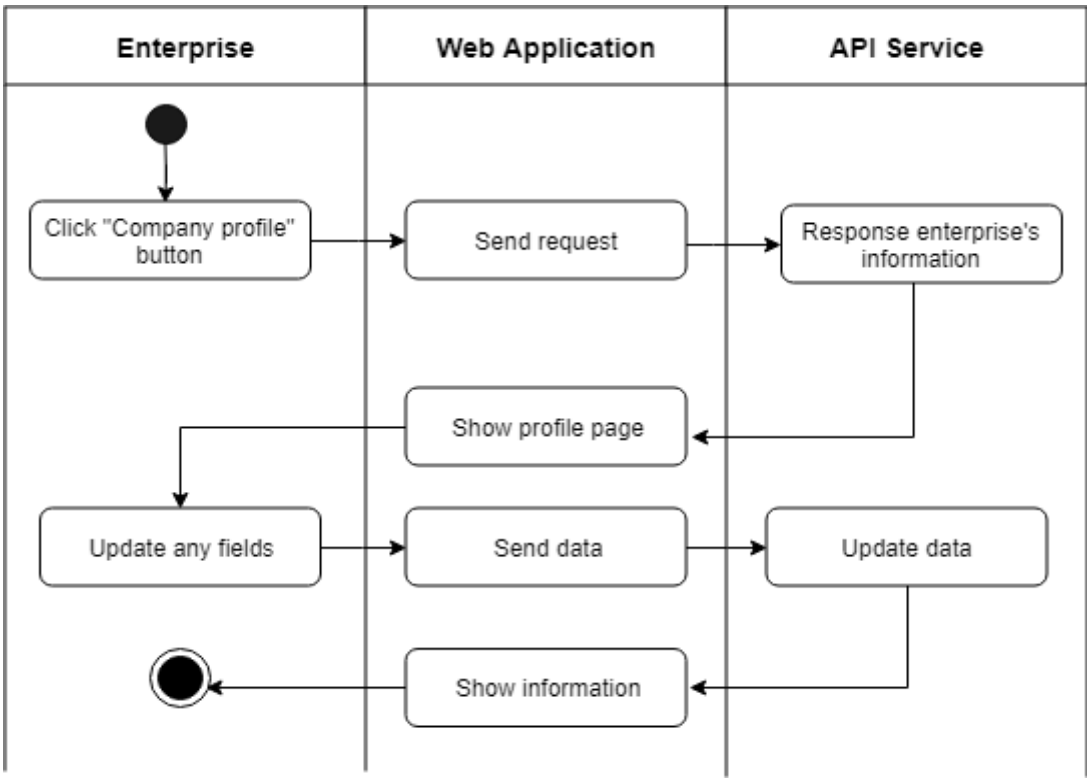

Figure 2.14 Activity diagram – Update enterprise's profile

10) Update student's profile

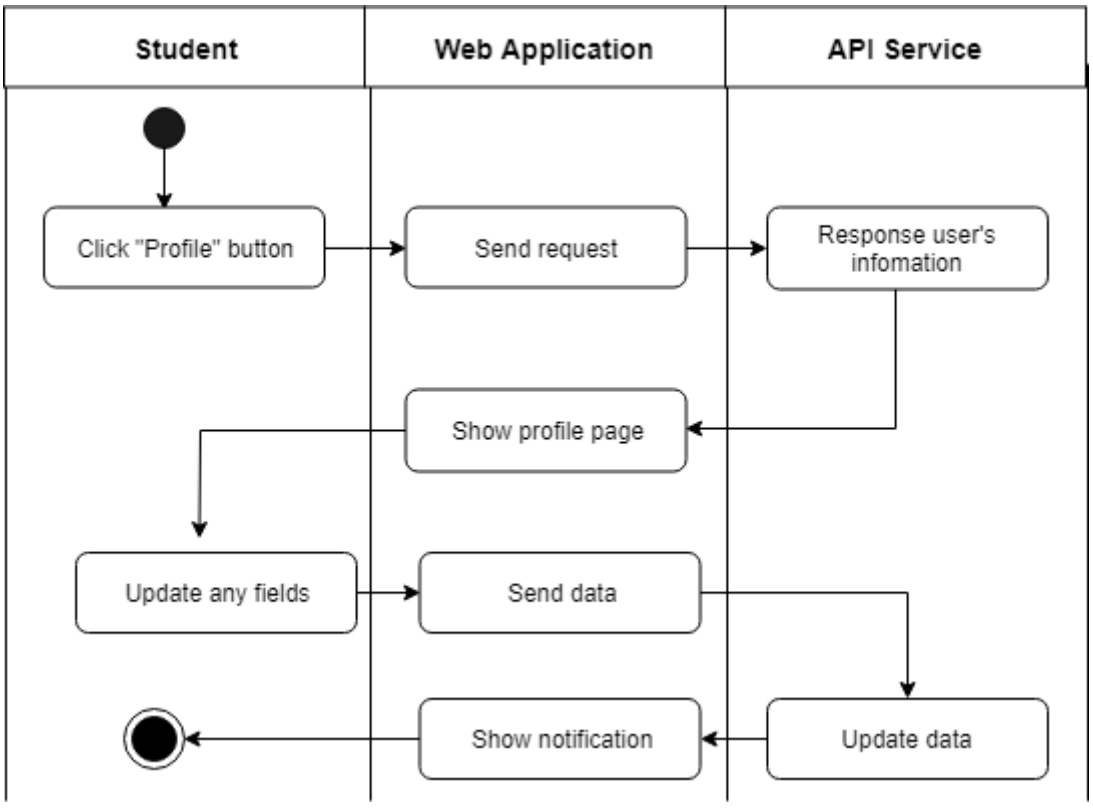

Figure 2.15 Activity diagram – Update student's profile

2.3. System design

2.3.1. Sequence diagram

1) Login

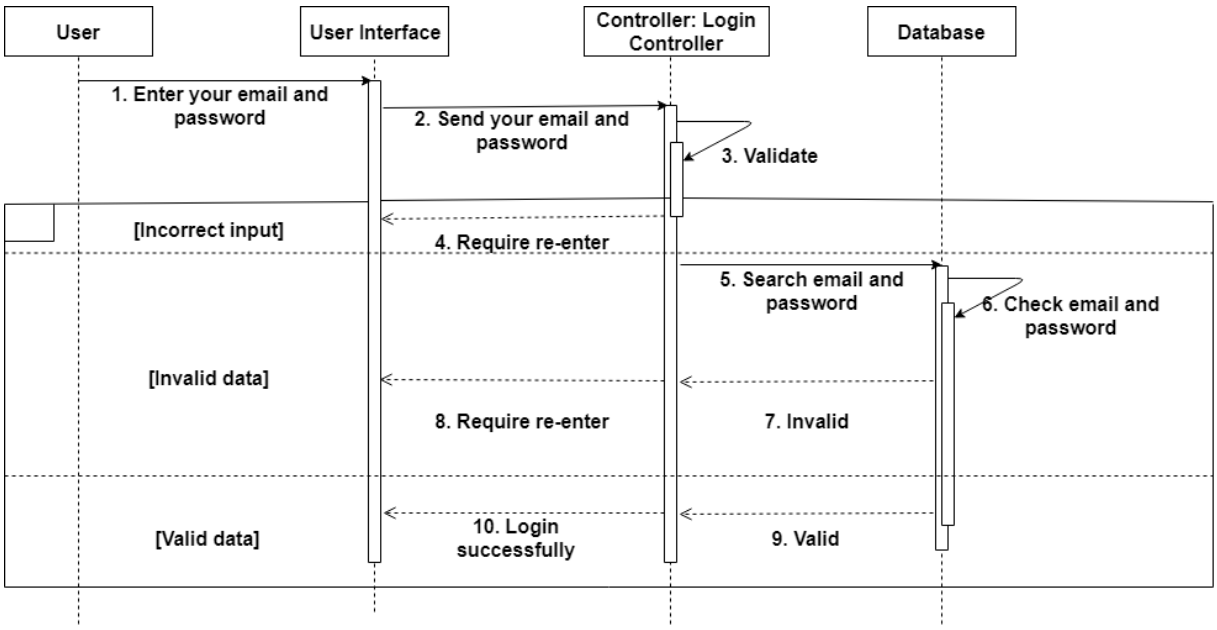

Figure 2.16 Sequence diagram – Login

2) Search jobs

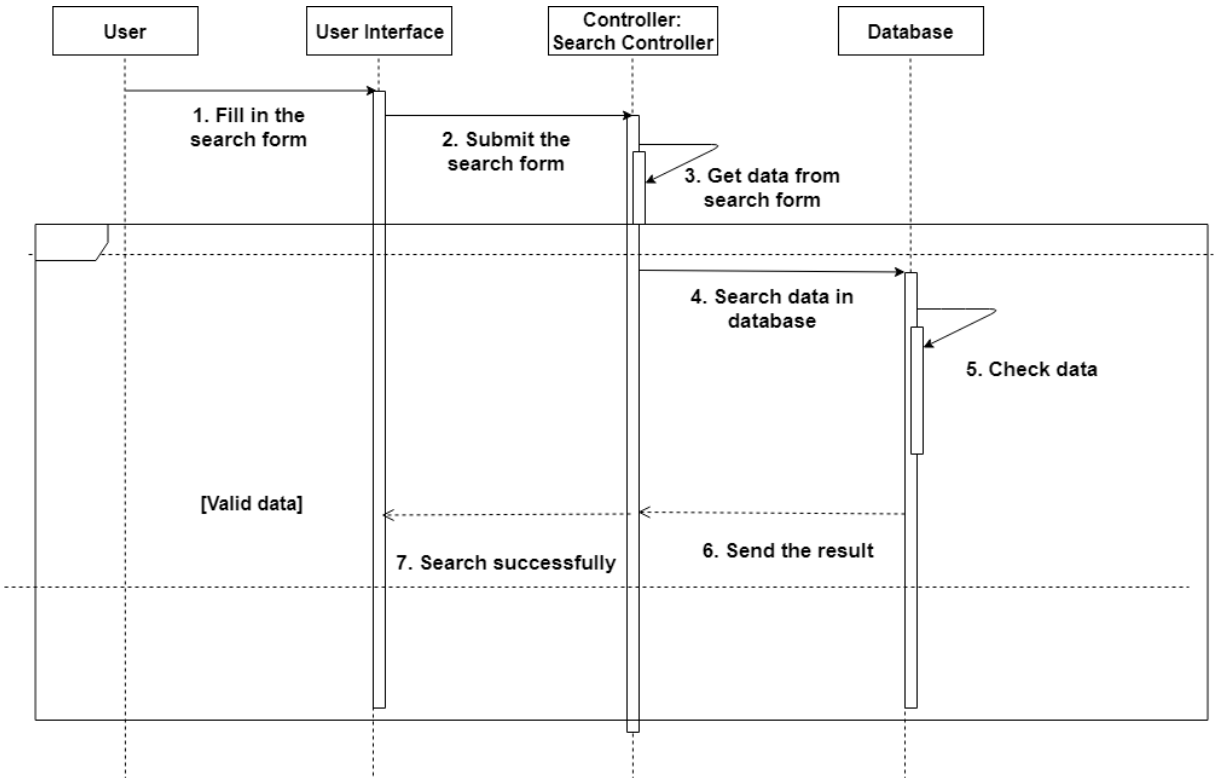

Figure 2.17 Sequence diagram – Search jobs

3) Apply for a job

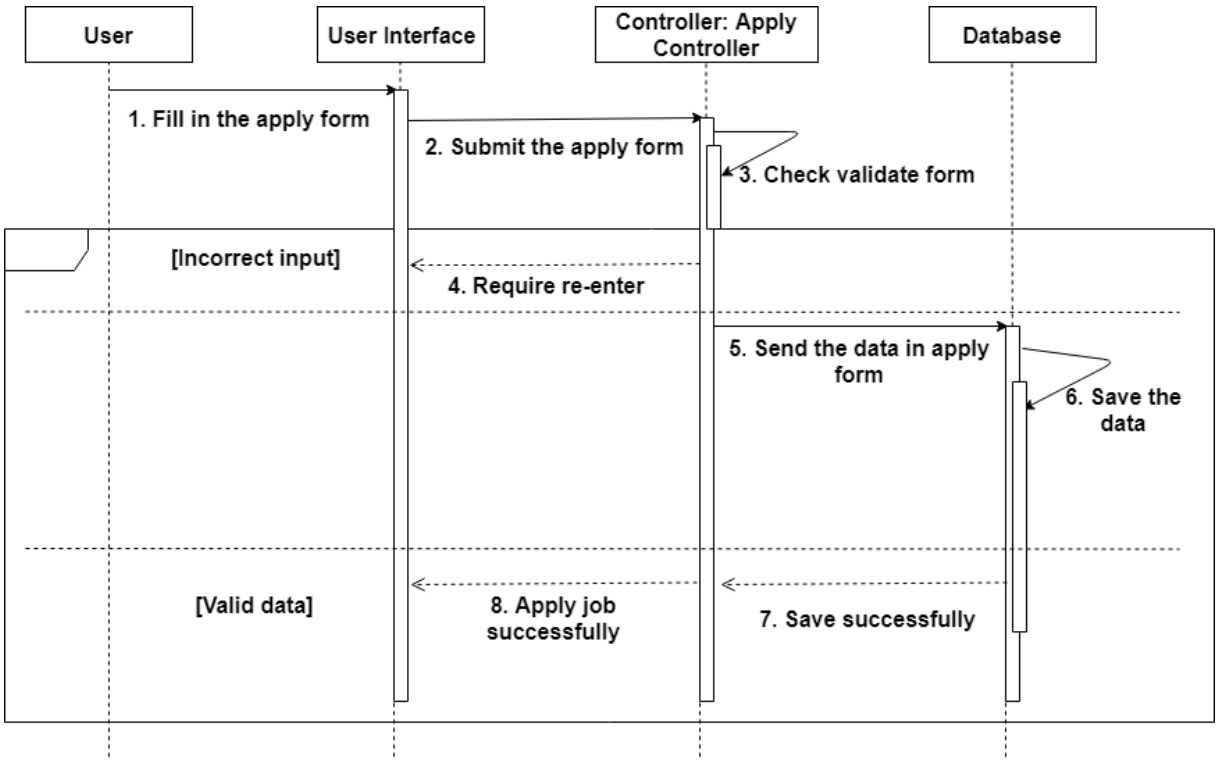

Figure 2.18 Sequence diagram – Apply for a job

5) Create a new job

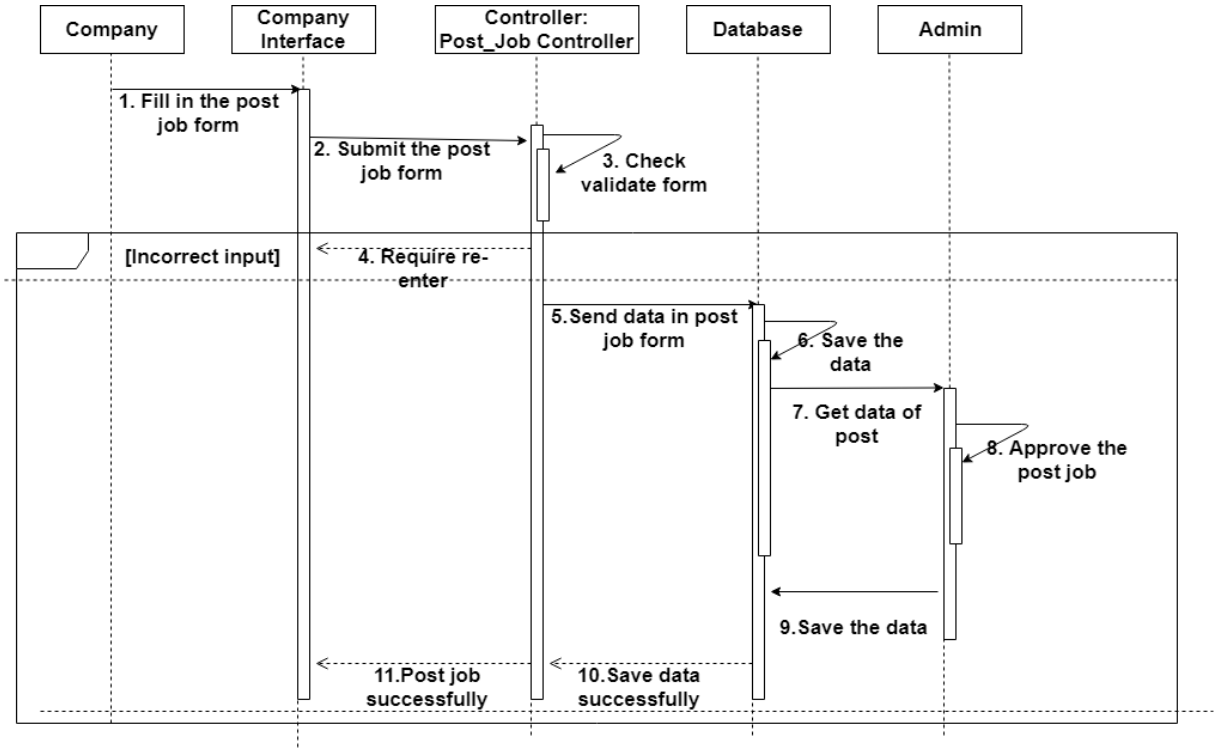

Figure 2.19 Sequence diagram – Create a new job

5) Create a new enterprise

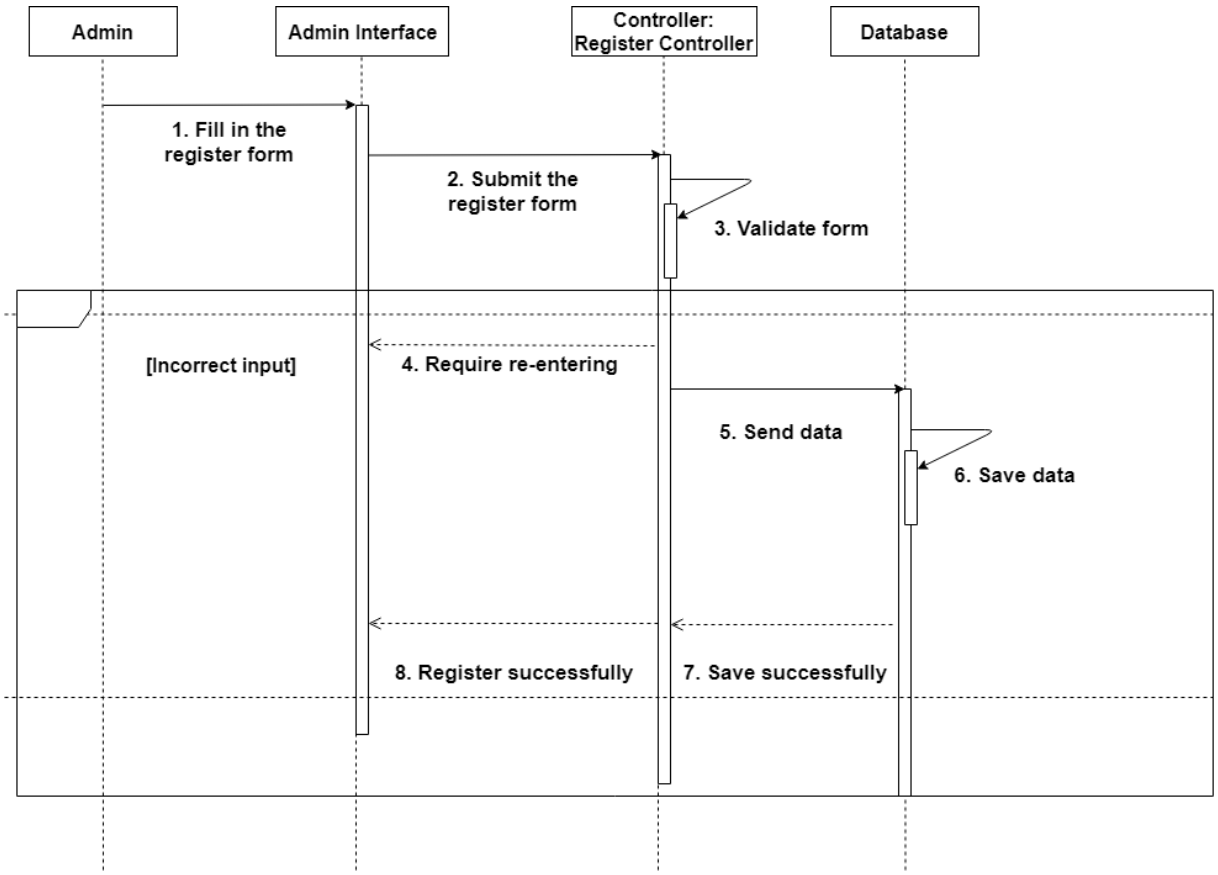

Figure 2.20 Sequence diagram – Create a new enterprise

2.3.2. Class diagram

The class diagram describes the responsibilities of a system, basis for components and deployment diagrams, analysis and design of the static view of an application.

It describes the relationship among all entities of this system, the attributes and methods are clearly identified.

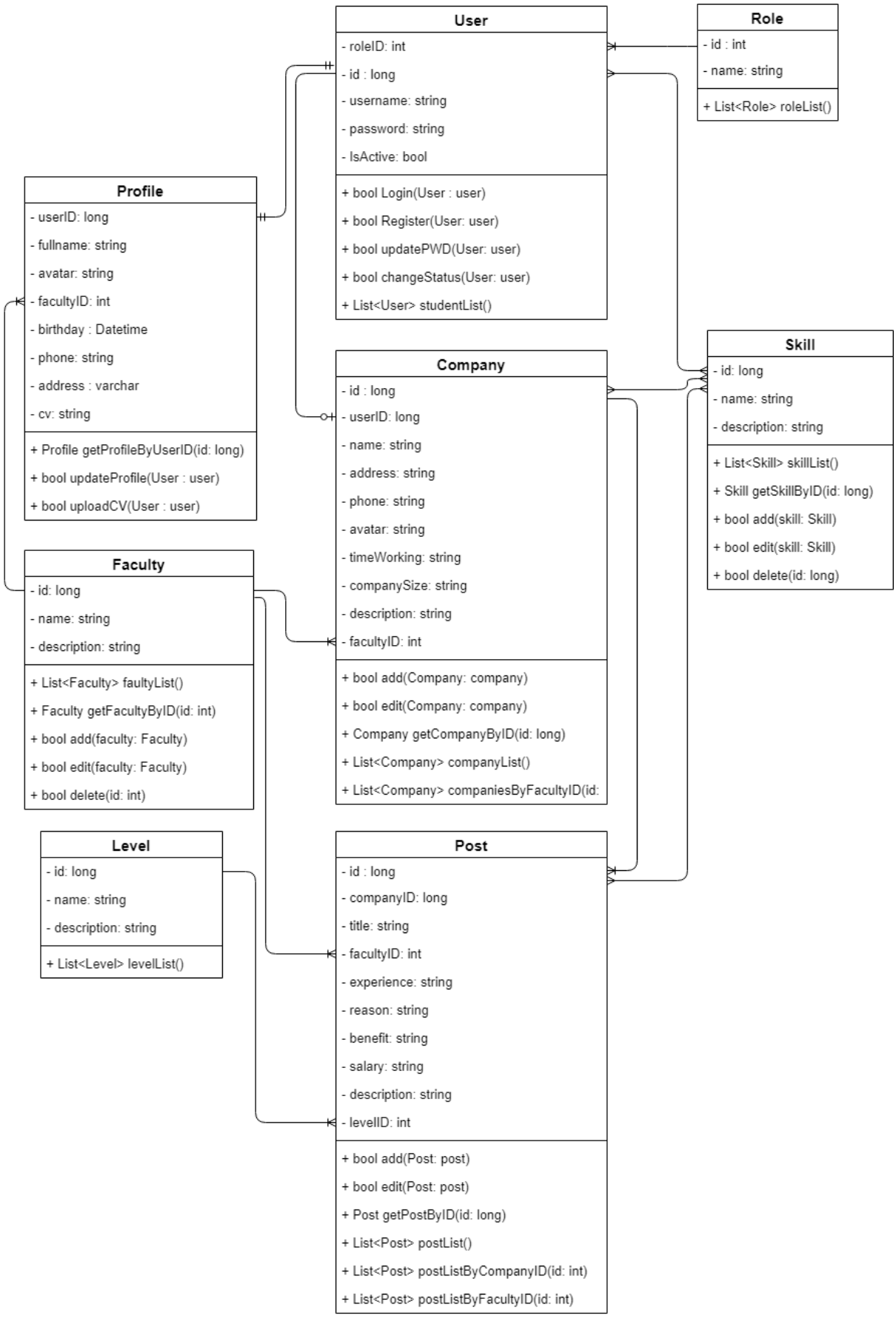

Figure 2.21 Class diagram

2.3.3. Database design

1) Relationship between tables

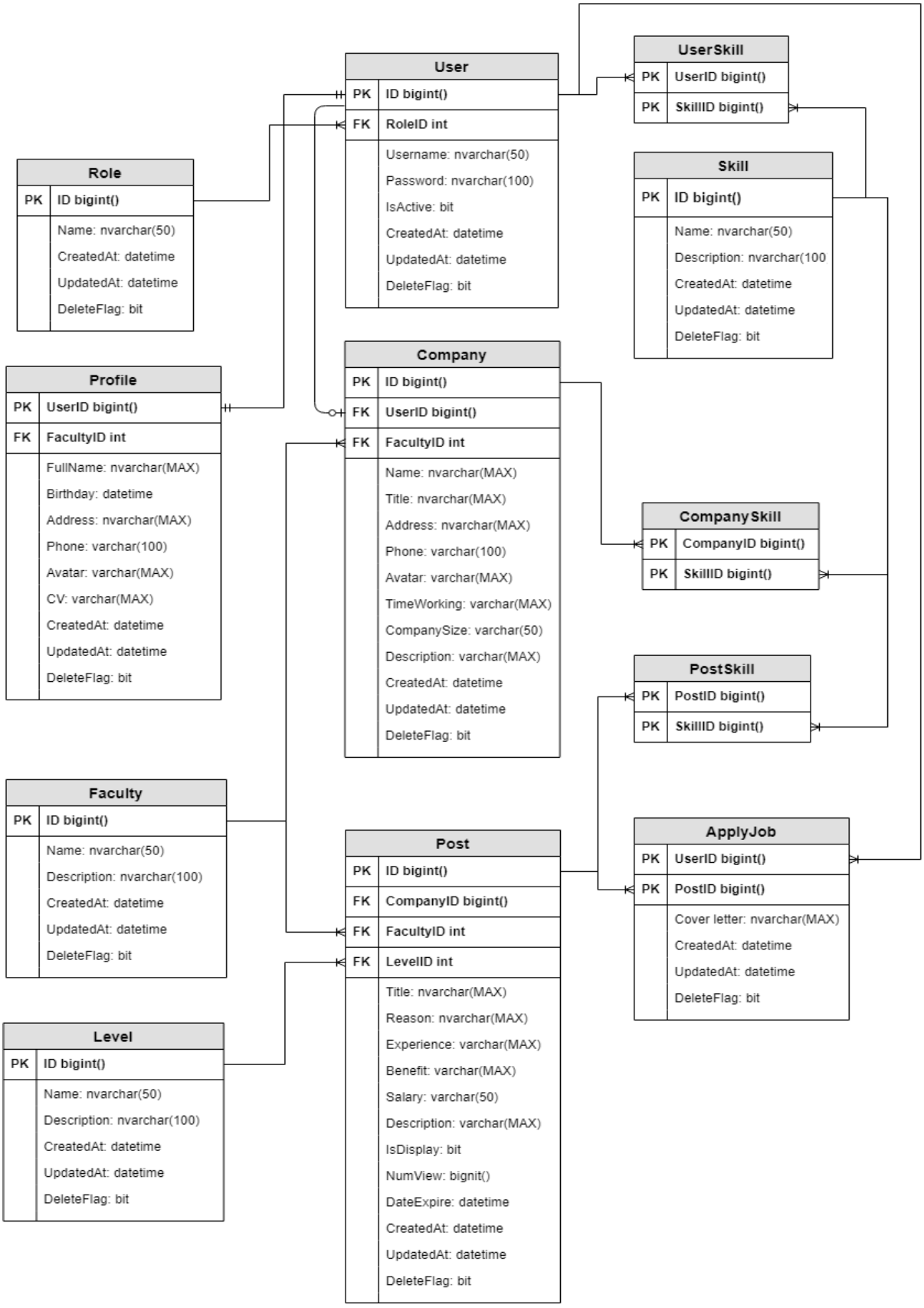

Figure 2.22 Database design

2) Description of tables

Table 2.9 Description of table – Role

No.	Name	Type	Length	Allow Null	Key
1	ID	int		FALSE	PK
2	Name	nvarchar	50	FALSE	
3	CreatedAt	datetime		FALSE	
4	UpdatedAt	datetime		FALSE	
5	DeleteFlag	bit		FALSE	

This table describes for roles of the system: admin, enterprise, student, guest.

Table 2.10 Description of table – User

No.	Name	Type	Length	Allow Null	Key
1	ID	bigint		FALSE	PK
2	Username	nvarchar	50	FALSE	
3	Password	nvarchar	100	FALSE	
4	IsActive	bit		FALSE	
5	CreatedAt	datetime		FALSE	
6	UpdatedAt	datetime		FALSE	
7	DeleteFlag	bit		FALSE	

This table stores login information for the entire system. Include basic information such as username, password, etc. It is linked to the role table via foreign keys “RoleID” to determine the role of each user.

Table 2.11 Description of table – Profile

No.	Name	Type	Length	Allow Null	Key
1	UserID	bigint		FALSE	PK
2	FacultyID	int	50	FALSE	FK
3	Fullname	nvarchar	100	FALSE	
4	Birthday	datetime		TRUE	
6	Address	nvarchar	MAX	TRUE	
7	Phone	nvarchar	100	TRUE	
8	Avatar	nvarchar	MAX	TRUE	
9	CV	nvarchar	MAX	TRUE	
10	CreatedAt	datetime		FALSE	

11	UpdatedAt	datetime		FALSE	
12	DeleteFlag	bit		FALSE	

This table contains detailed personal information for each User. Each a User (Table 2.9) has to a Profile (Table 2.10).

Table 2.12 Description of table – Company

No.	Name	Type	Length	Allow Null	Key
1	ID	bigint		FALSE	PK
2	UserID	bigint		FALSE	FK
3	FacultyID	int		FALSE	FK
4	Name	nvarchar	100	FALSE	
5	Title	nvarchar	MAX	FALSE	
6	Address	nvarchar	MAX	FALSE	
7	Phone	nvarchar	100	FALSE	
8	Avatar	nvarchar	MAX	TRUE	
9	TimeWorking	nvarchar	100	FALSE	
10	CompanySize	nvarchar	100	FALSE	
11	Description	nvarchar	MAX	FALSE	
12	CreatedAt	datetime		FALSE	
13	UpdatedAt	datetime		FALSE	
14	DeleteFlag	bit		FALSE	

This table stores information about companies. Each company will be managed by a User (Table 2.10).

Table 2.13 Description of table – Faculty

No.	Name	Type	Length	Allow Null	Key
1	ID	bigint		FALSE	PK
2	Name	nvarchar	50	FALSE	
3	Avatar	nvarchar	MAX	TRUE	
4	Description	nvarchar	100	TRUE	
5	CreatedAt	datetime		FALSE	
6	UpdatedAt	datetime		FALSE	
7	DeleteFlag	bit		FALSE	

This table describes the faculties of DUT such as Information Technology, Electrical, Chemical Sciences, etc.

Table 2.14 Description of table – Level

No.	Name	Type	Length	Allow Null	Key
1	ID	bigint		FALSE	PK
2	Name	nvarchar	50	FALSE	
3	Description	nvarchar	100	TRUE	
4	CreatedAt	datetime		FALSE	
5	UpdatedAt	datetime		FALSE	
6	DeleteFlag	bit		FALSE	

This table describes the level of a hiring post such as internship, fresher, etc.

Table 2.15 Description of table – Skill

No.	Name	Type	Length	Allow Null	Key
1	ID	bigint		FALSE	PK
2	Name	nvarchar	50	FALSE	
3	Description	nvarchar	100	TRUE	
4	CreatedAt	datetime		FALSE	
5	UpdatedAt	datetime		FALSE	
6	DeleteFlag	bit		FALSE	

This table describes skills such as PHP, Java, Angular, etc.

Table 2.16 Description of table – Post

No.	Name	Type	Length	Allow Null	Key
1	ID	bigint		FALSE	PK
2	CompanyID	bigint		FALSE	
3	FacultyID	int		FALSE	
4	LevelID	int		FALSE	
5	Title	nvarchar	MAX	FALSE	
6	Reason	nvarchar	MAX	FALSE	
7	Experience	nvarchar	MAX	FALSE	
8	Benefit	nvarchar	MAX	FALSE	
9	Salary	nvarchar	MAX	FALSE	
10	Description	nvarchar	MAX	FALSE	
11	IsDisplay	bit		FALSE	
12	NumView	int		FALSE	
13	DateExpire	datetime		FALSE	

14	CreatedAt	datetime		FALSE	
15	UpdatedAt	datetime		FALSE	
16	DeleteFlag	bit		FALSE	

This table describes the details of the recruiting posts.

Table 2.17 Description of table – UserSkill

No.	Name	Type	Length	Allow Null	Key
1	UserID	bigint		FALSE	PK
2	SkillID	bigint		FALSE	PK

Table 2.18 Description of table – CompanySkill

No.	Name	Type	Length	Allow Null	Key
1	UserID	bigint		FALSE	PK
2	SkillID	bigint		FALSE	PK

Table 2.19 Description of table – PostSkill

No.	Name	Type	Length	Allow Null	Key
1	UserID	bigint		FALSE	PK
2	SkillID	bigint		FALSE	PK

Table 2.20 Description of table – ApplyJob

No.	Name	Type	Length	Allow Null	Key
1	UserID	bigint		FALSE	PK
2	PostID	bigint		FALSE	PK
3	CoverLetter	nvarchar	MAX	TRUE	
4	CreatedAt	datetime		FALSE	
5	UpdateAt	datetime		FALSE	
6	DeleteFlag	bit		FALSE	

This table is used to store information of candidates who apply for jobs of enterprises.

2.4. Conclusions

This chapter presented the requirements specification that the system could meet the user' demands. Follow the requirements, the use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. The activity diagrams, draw the activity flow of a system, show the steps and action sequences as well as the interactions between user and user, user and system. Thereby, the overview and the activity streams of the system are fully presented.

Besides, it also describes the system structure, as well as the action sequences for each function. By that, it facilitates the testing phase, the tester can go back the sequence diagrams to follow the action sequences and create the function tests and the input data as well. Furthermore, it shows the database and some UI application designs to help the reader have clearer views of the system. By that, I can consider and evaluate the complexity of this system.

Chapter 3 : IMPLEMENTATION AND EVALUATION

3.1. Development environment

- **Web service**

The website service is a RESTful API built by .NET Core 2.1 Framework. I use SQL Server to store data.

- **Web client**

I use Angular version 7 to develop my website application.

- **Software Development Tools**

I use tools: Microsoft Visual Studio, Visual Studio Code and Postman.

3.2. Data collection

For having initial data, I chose to crawl data from some websites such as itviec.com [6], vietnamwork.com [7]. I have collected more than 1200 job records, enterprise's information as well as job skills from them. After I finished collecting, I classified the data by faculties and labeled based on skills and levels. Finally, the data will be stored in my database.

3.3. System demo

3.3.1. Login page

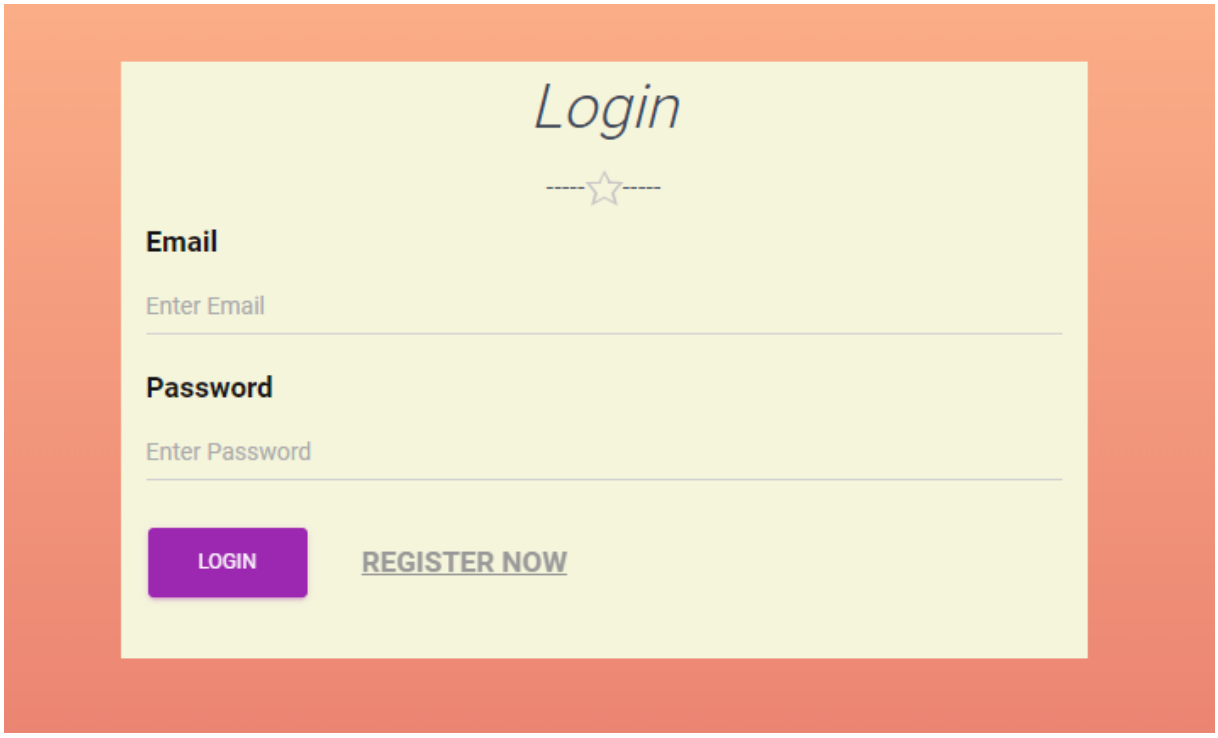

Figure 3.1 Login page

The website uses email and password to log in. After login successful, it will direct the user who is a student or an enterprise to the index page. For user is admin will be redirected to the dashboard page.

3.3.2. Index page

The index page is acceptable for all users to access. Include users who are logged in or not. At here displayed in order from top to bottom: the first is the header, the second is the form to search, the third is the overview of the application, the continue is 4 faculties have the most hiring posts, after that is 8 latest hiring posts and the last is the footer of the system.

Figure 3.2 The header of the index page

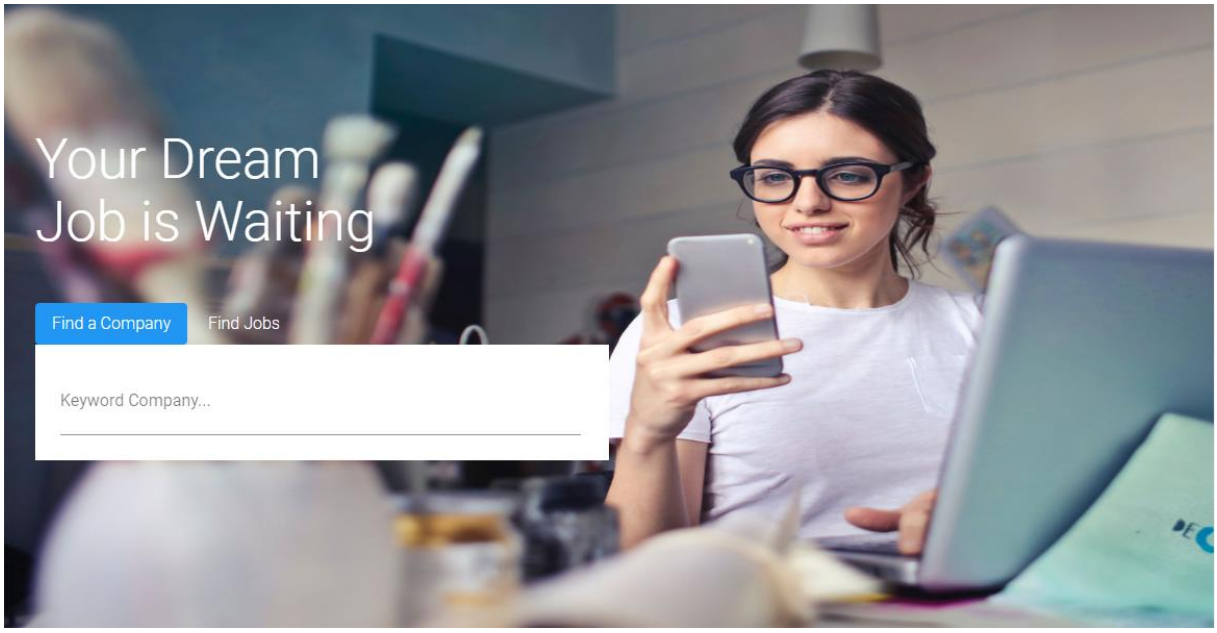

Figure 3.3 The search form of index page

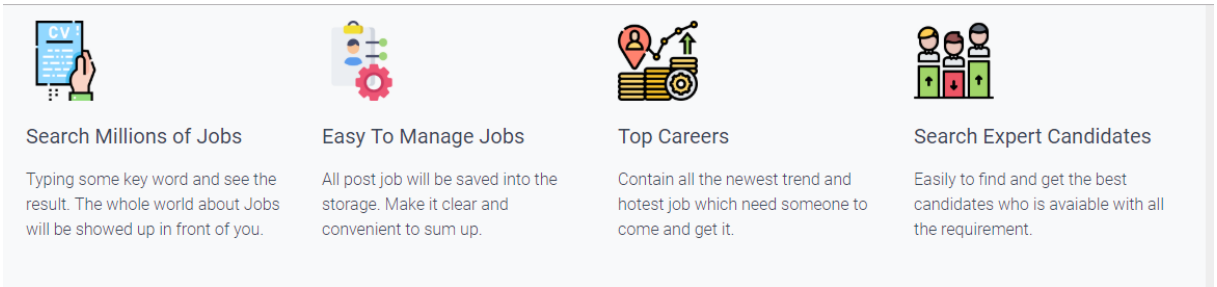

Figure 3.4 The overview about the application of the index page

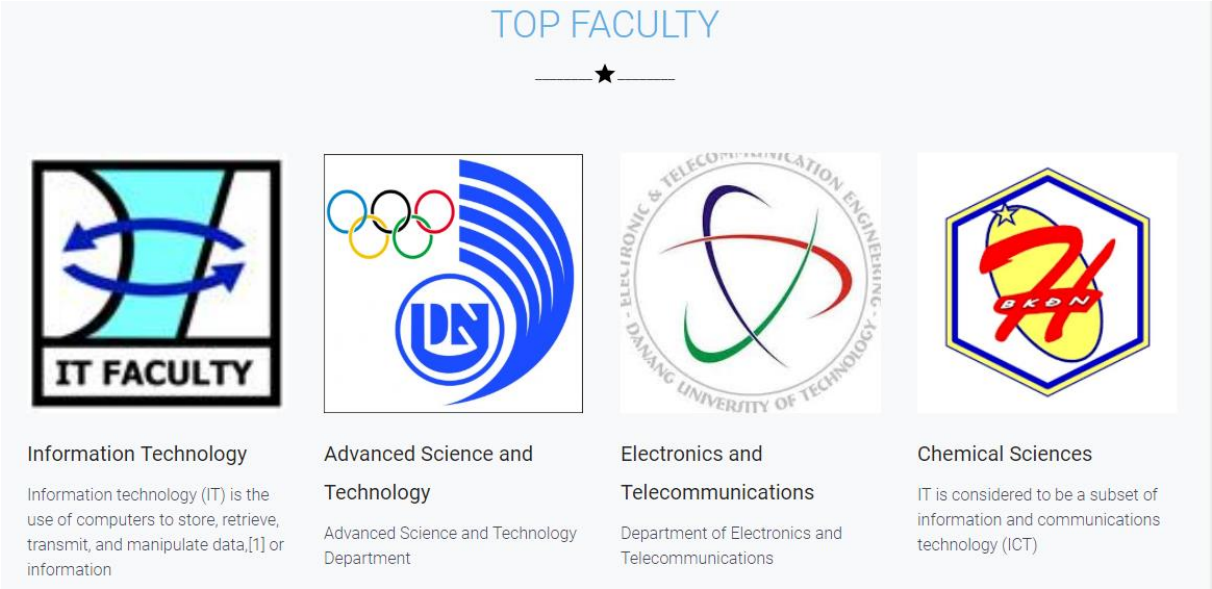

Figure 3.5 Top 4 faculties have the most hiring posts of the index page

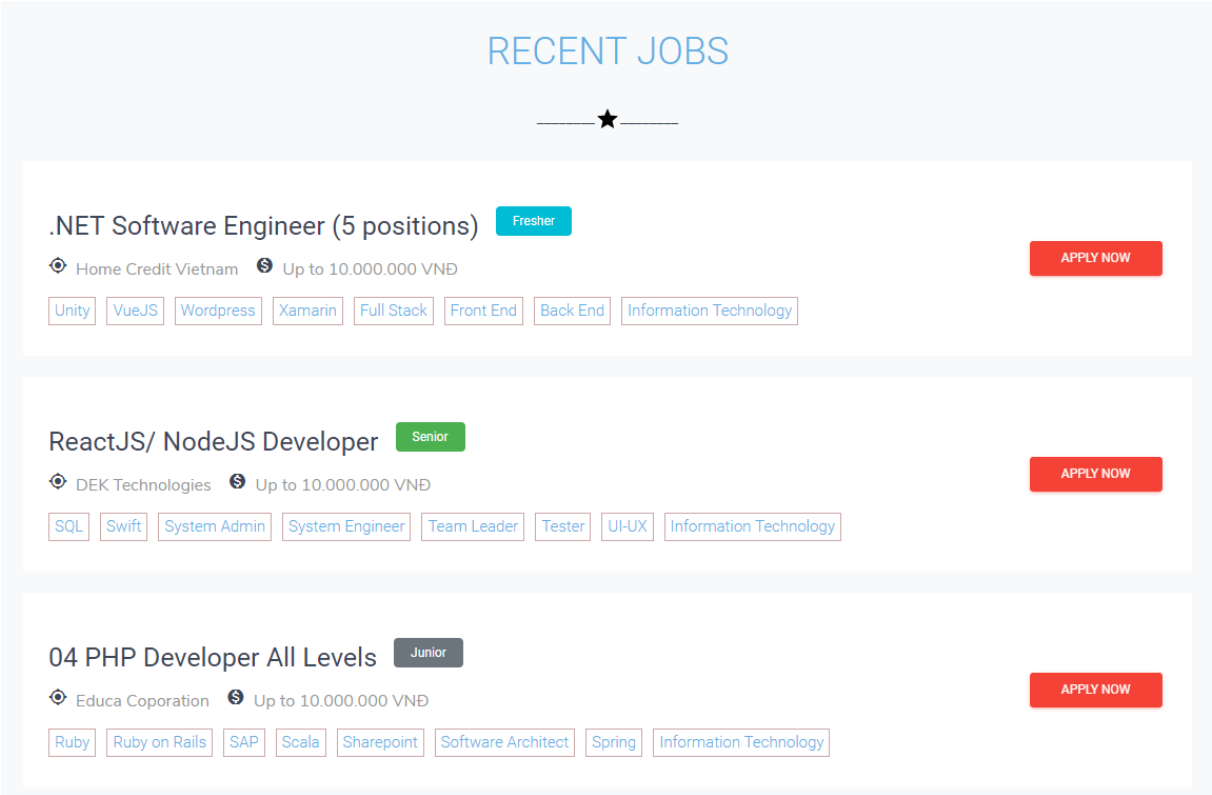

Figure 3.6 Top 8 latest hiring posts of the index page

Figure 3.7 The footer of the index page

3.3.3. Enterprise page

At the search form of the index page, select the tab Find a Company and enter a name. The system will suggest a few companies fit for your keyword. Click a name which you want to see. The website will navigate to the index page.

For example, I enter a name is ***En***. The system suggested a company name which is ***Enclave***. Click this name to redirect to the enterprise page.

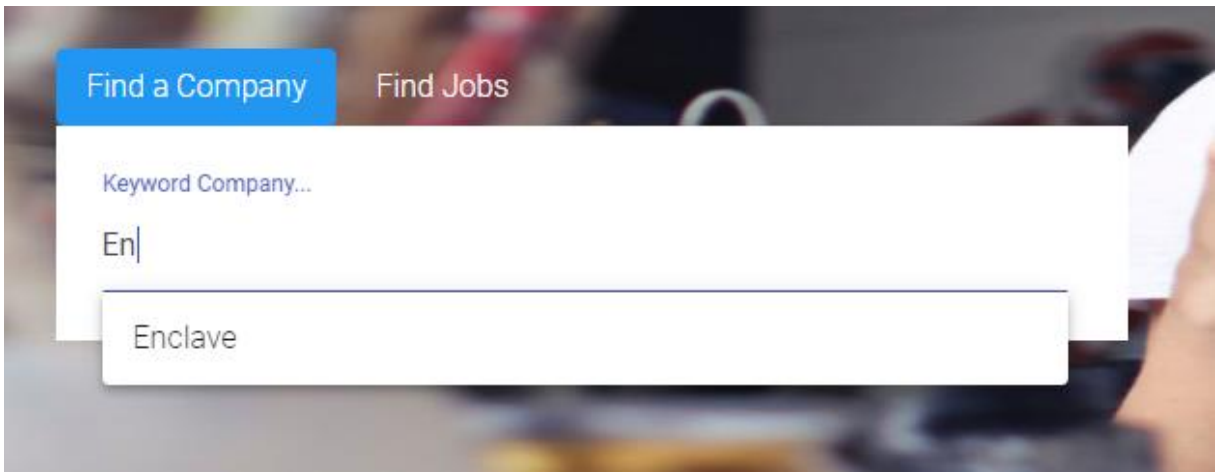

Figure 3.8 Enter keyword to search a company

The enterprise page has some information about the enterprise such as:

- Name
- Address
- Phone number
- Enterprise's overview
- Skills
- Jobs offered by its.
- Review

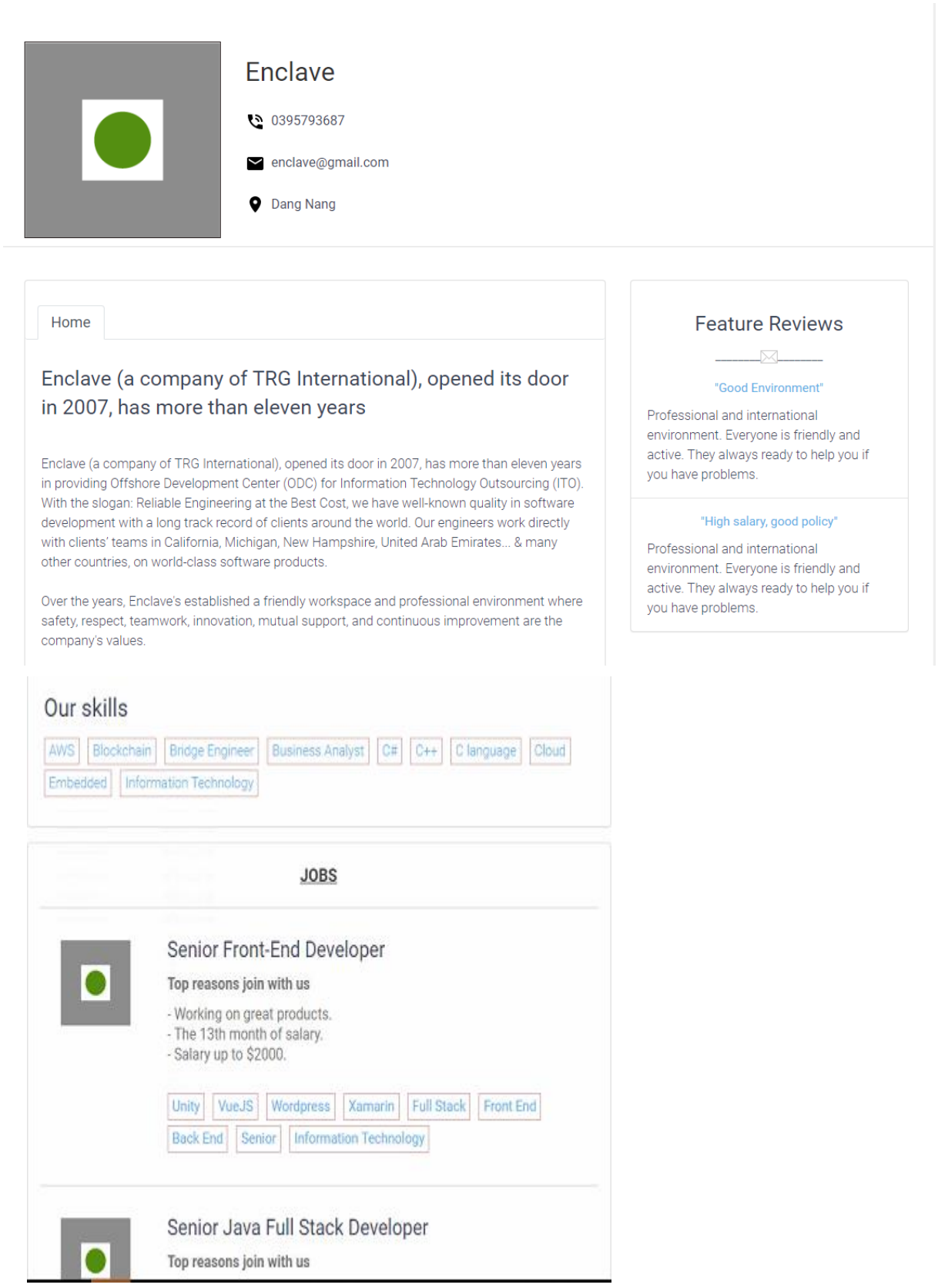

Figure 3.9 The information detail of the enterprise page

3.3.4. Searching result page

At the search form of the index page, select the Find jobs tab and enter a keyword, that is a skill, a faculty, etc.

If you want to search by a skill. For example, **PHP**. You can enter a keyword which is ***PHP***. Press Enter to redirect search result page.

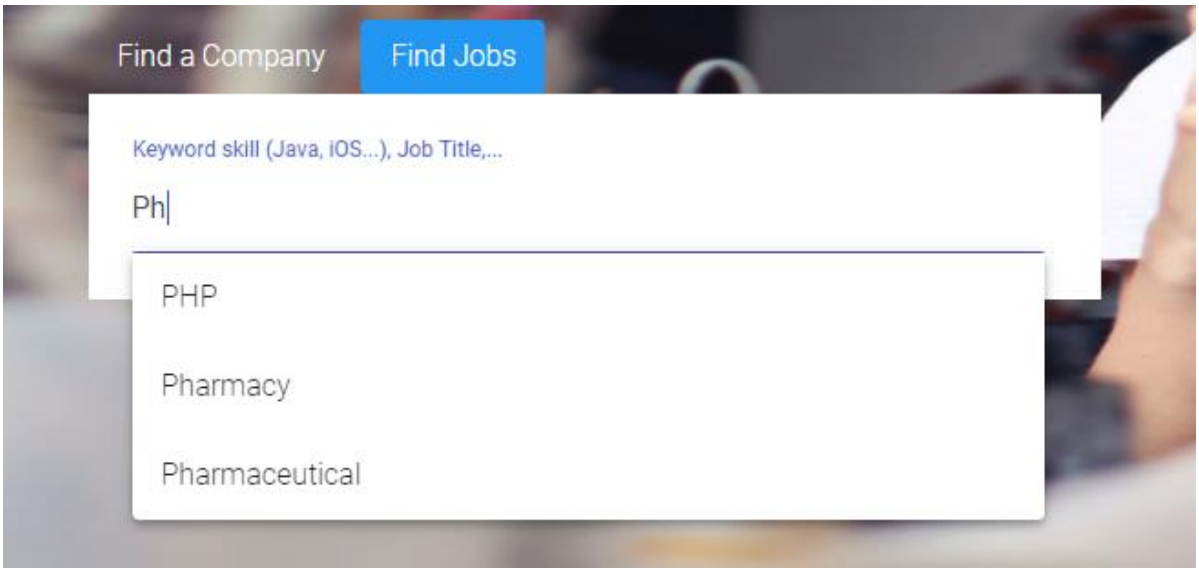

*Figure 3.10 Enter a skill is "**PHP**" to find jobs*

Search results are sorted by the latest hiring post.

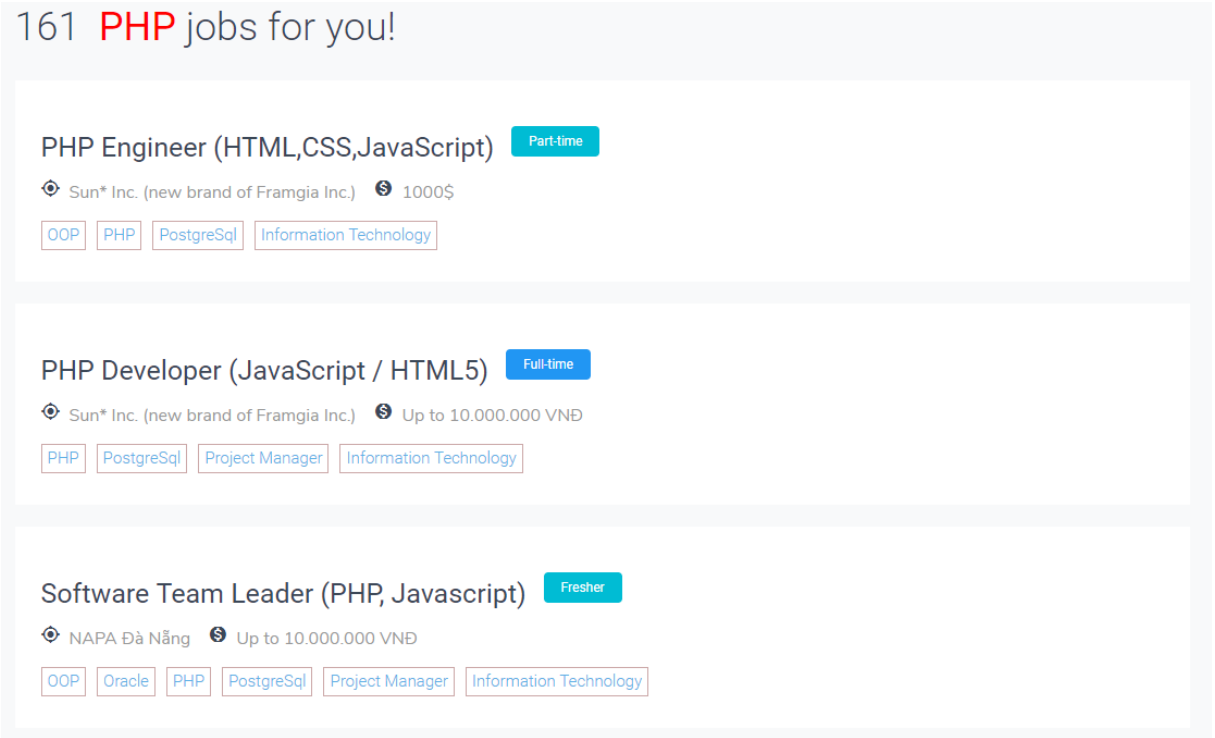

*Figure 3.11 Search result with the skill is "**PHP**"*

If the system can't find any job that matches your keyword, it still gives you for some jobs that are closest to the keyword you're looking for.

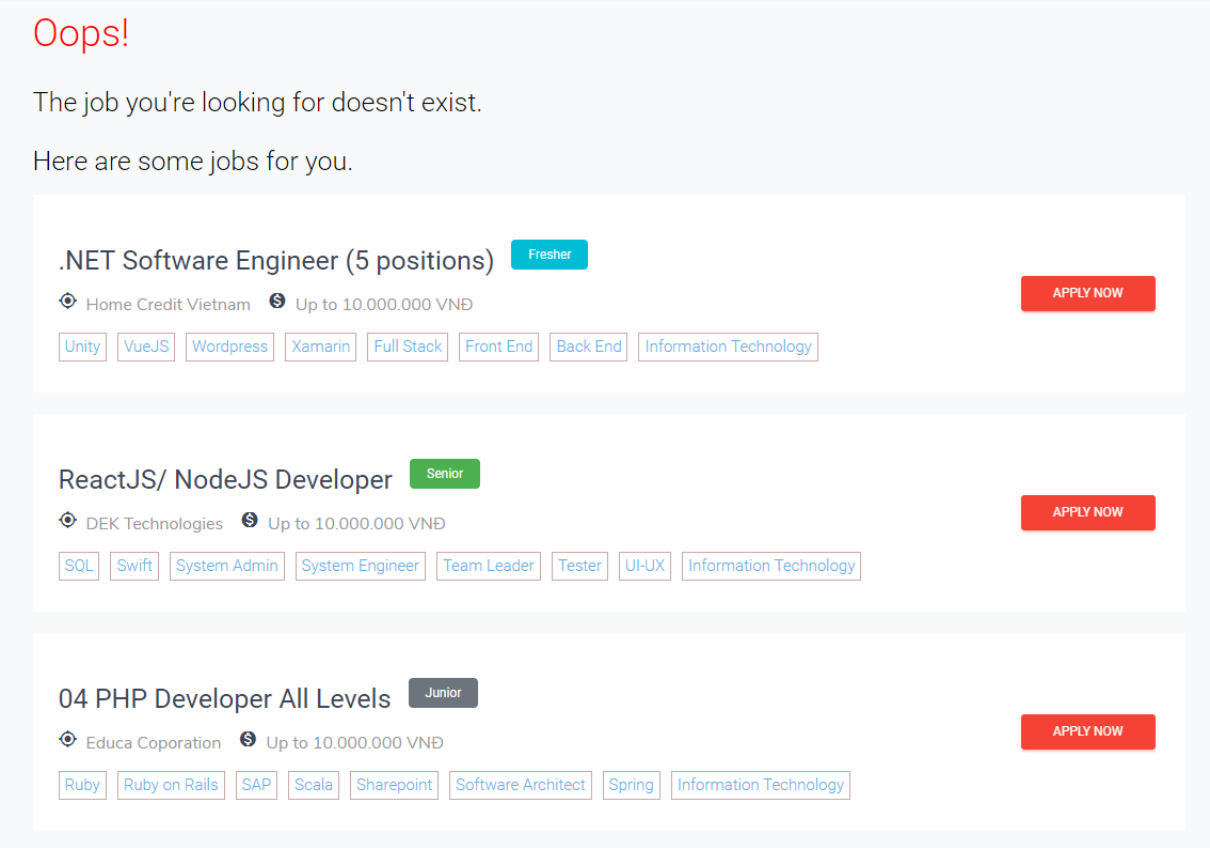

Figure 3.12 Search result closest the keyword

You also search by level. If you choose a keyword is "**Internship"** at the Find jobs tab. You can see all the **Internship** jobs at the searching result page.

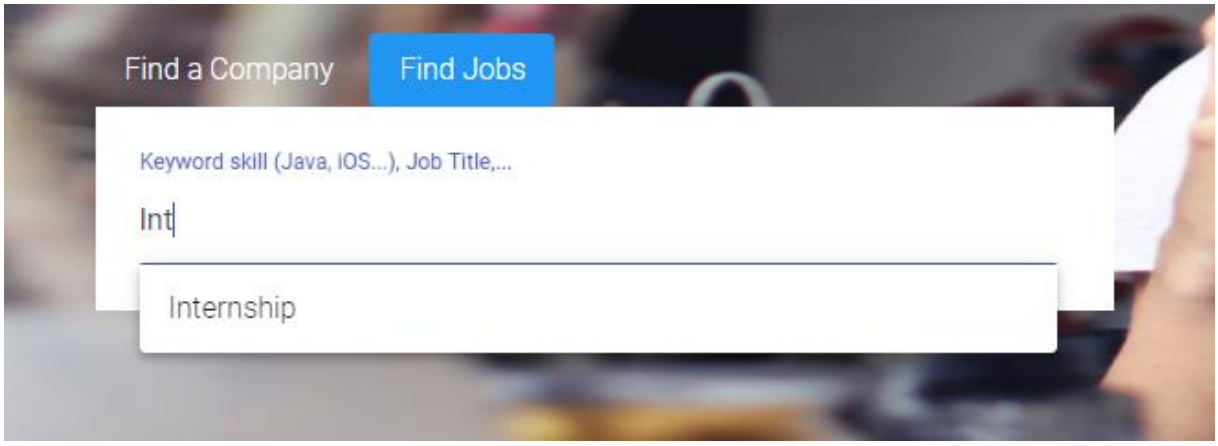

*Figure 3.13 Enter a level is "**Internship**" to find jobs*

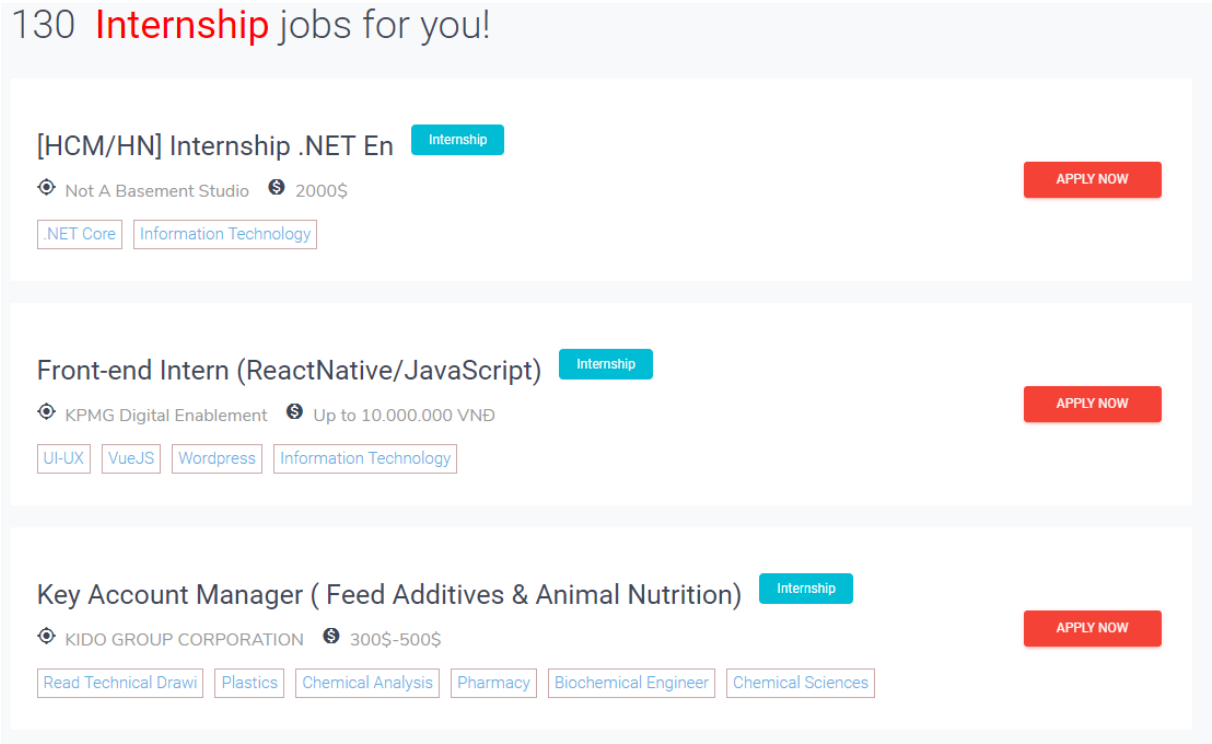

*Figure 3.14 Search result with the level is "**Internship**"*

If you want to view all the job provided by your faculty. You enter your faculty name in the Find jobs tab. The system will search and response for you that all of this faculty's job.

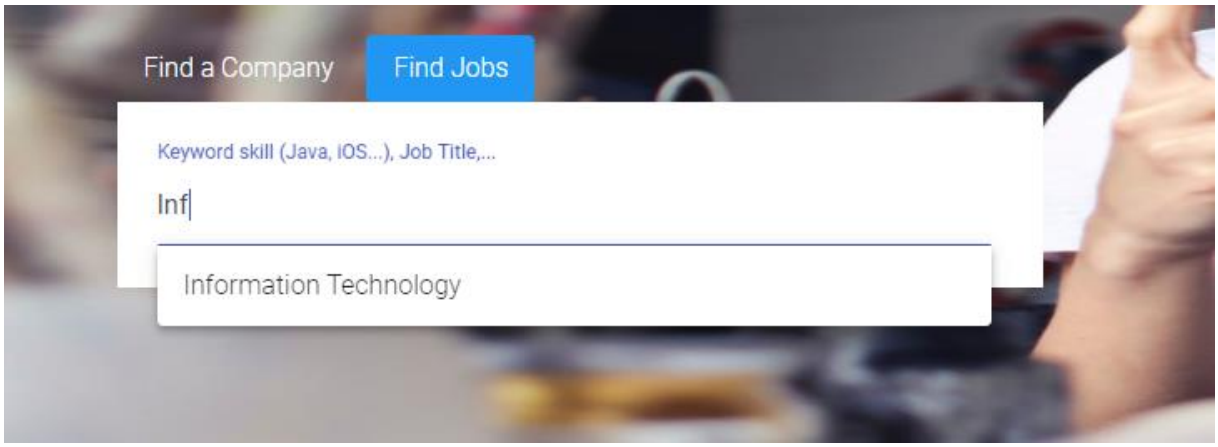

*Figure 3.15 Enter a faculty is "**Information Technology**" to find jobs*

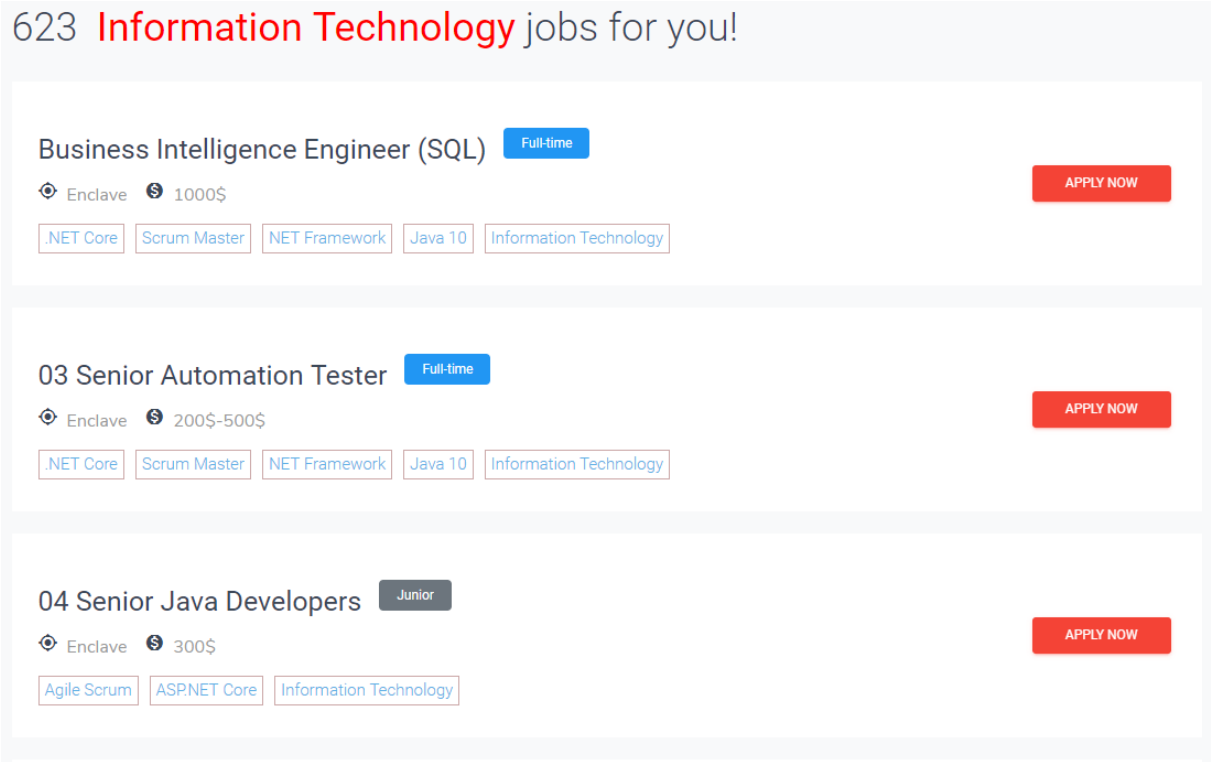

*Figure 3.16 Search result with the faculty is "**Information Technology**"*

3.3.5. Job page

At the search result page, you click on a job to see details. Details of a job page include the following information:

- Job's name
- Skills
- Salary
- Description
- Experience and skill needed for this job
- Information the company provides this job
- Some other jobs provided by this company

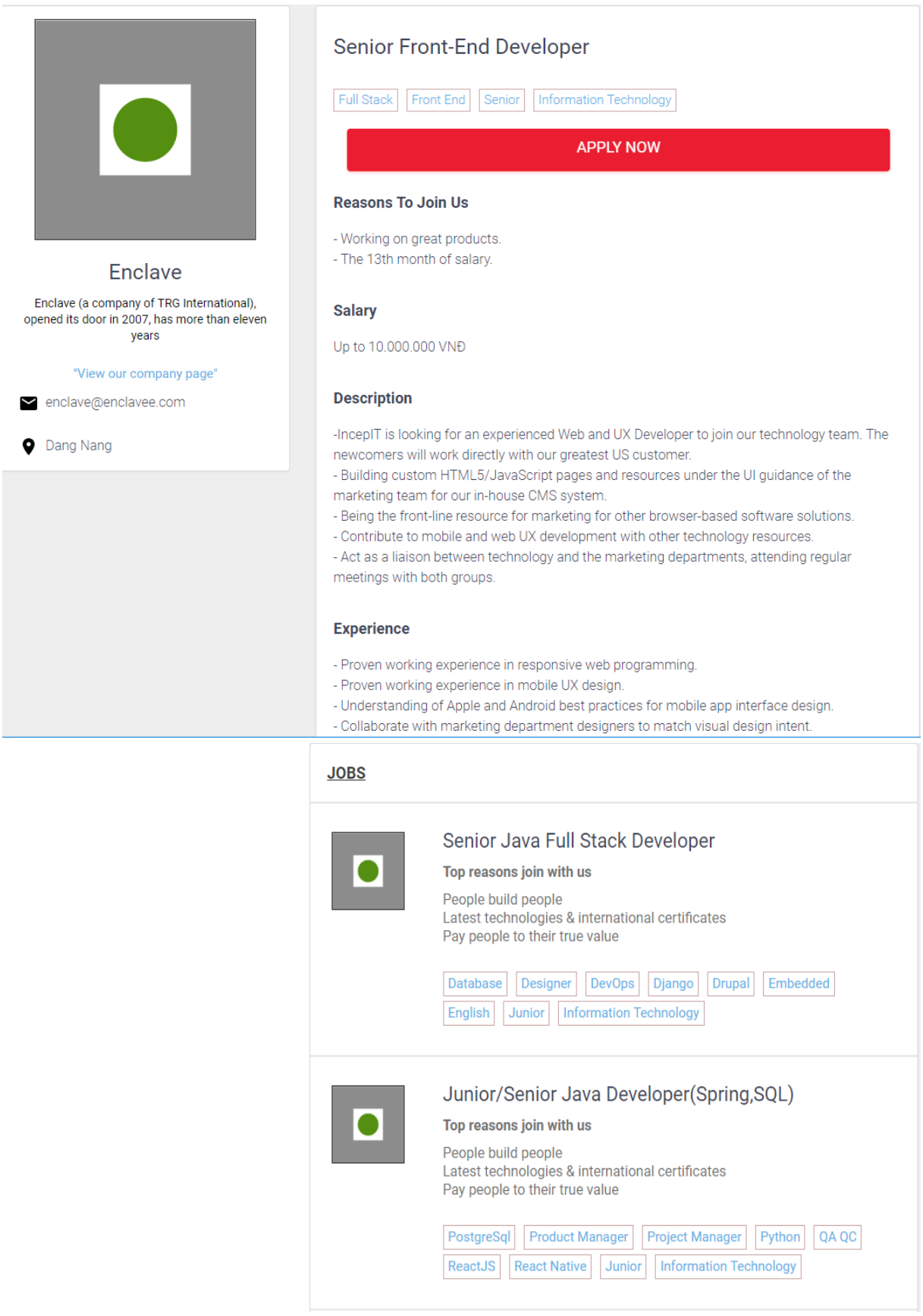

Figure 3.17 The detail job page

3.3.6. *Applying for a job*

You must be logged in to apply for a job. You should fill out the full form of the application so that employers have a good view of you. After you submit the application, we will send them to the email of this employer.

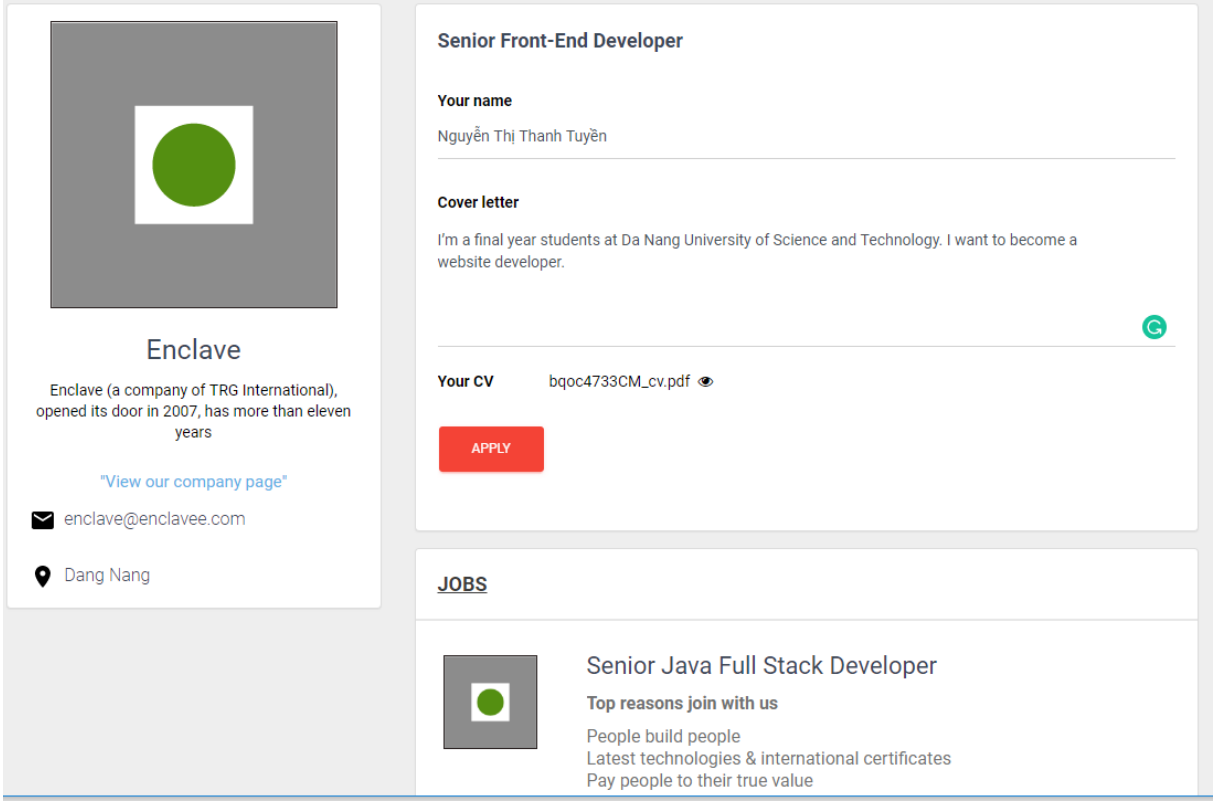

Figure 3.18 The application form

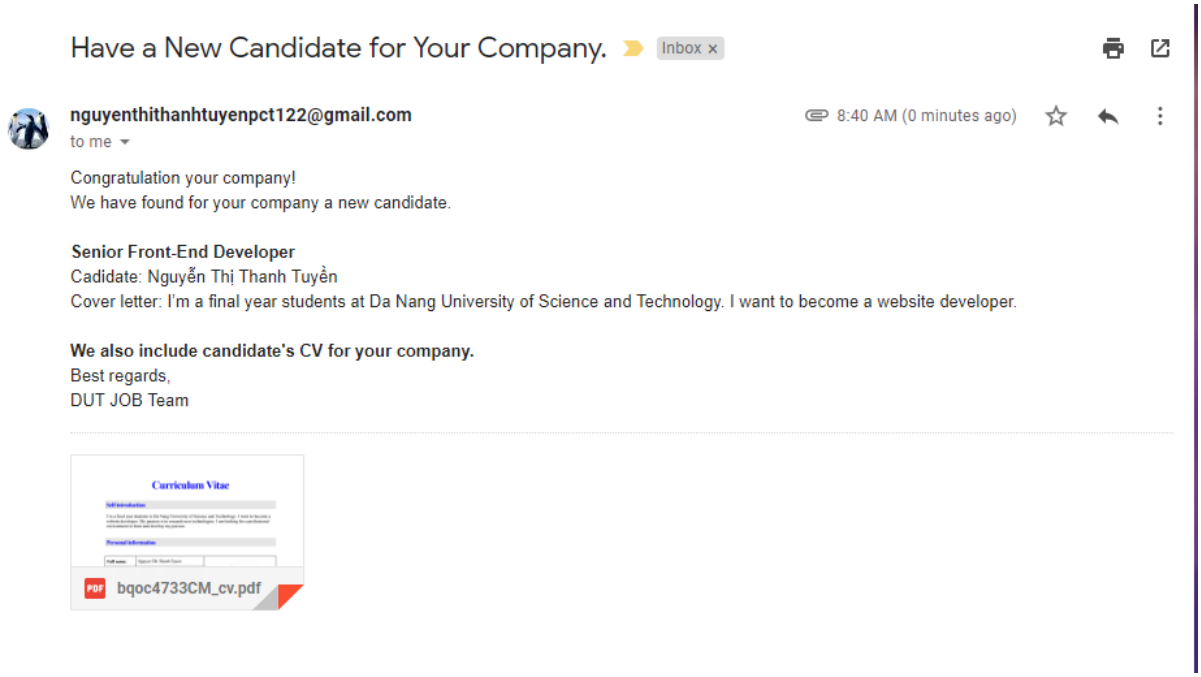

Figure 3.19 An email sent to the employer

3.3.7. Personal profile page

1) Student

At the personal profile page, the student can:

- Updating the profile
- Viewing the skill list
- Adding new skills
- Viewing the applied job list
- Viewing a CV
- Uploading a new CV

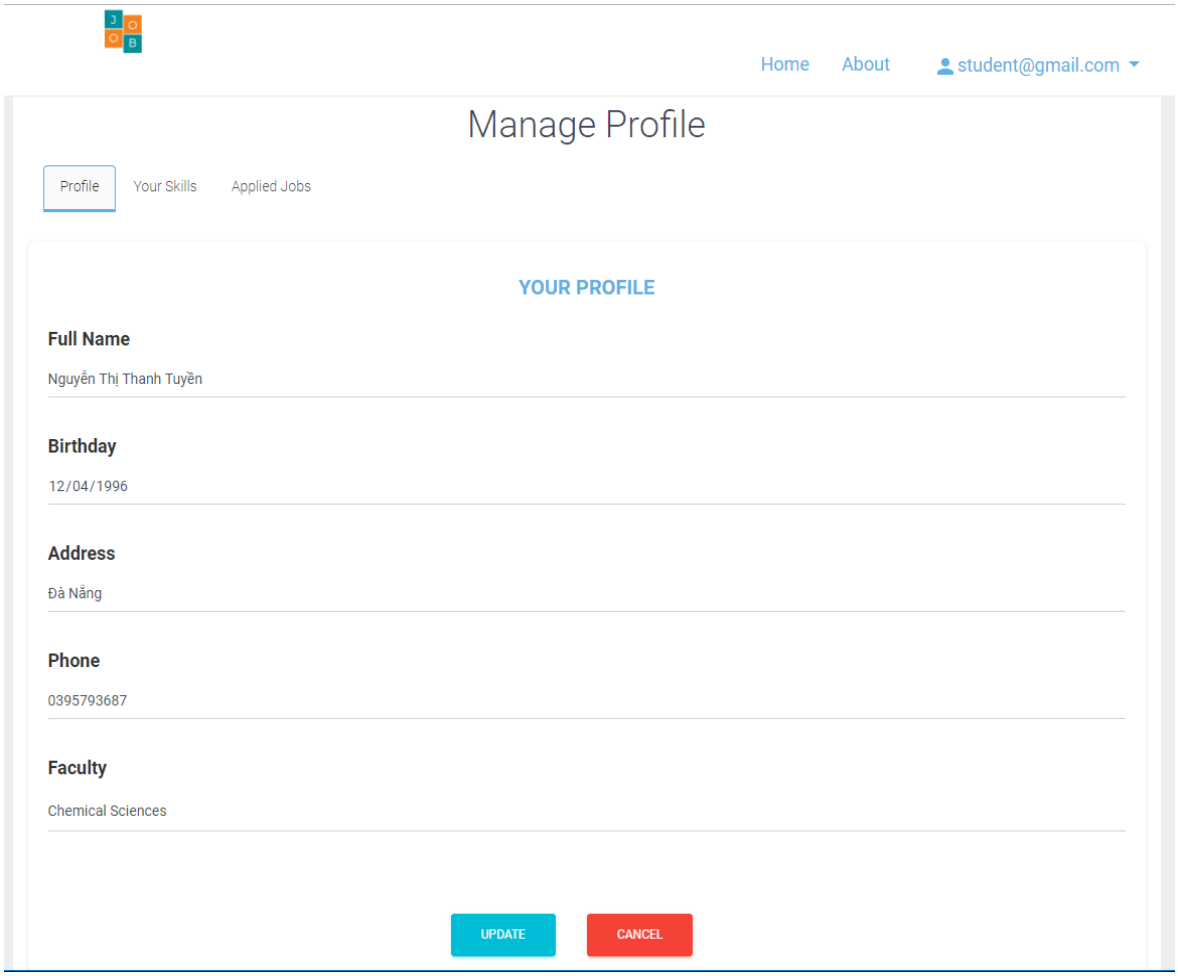

Figure 3.20 Your profile page

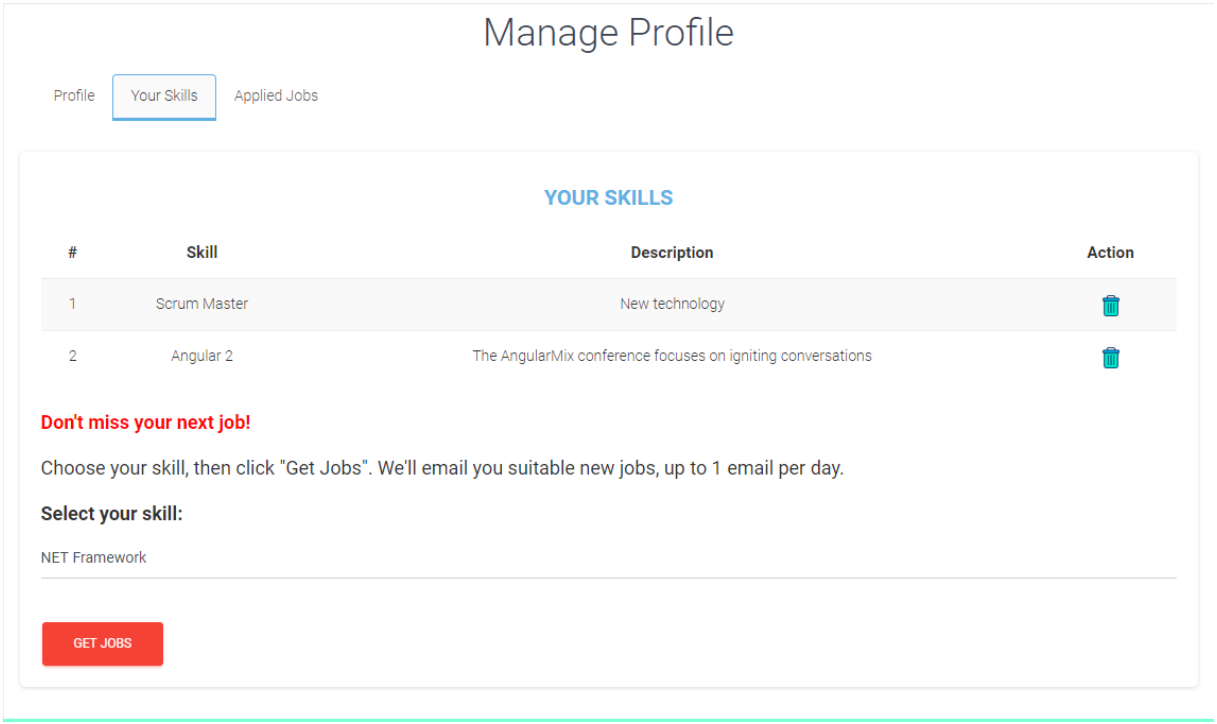

Figure 3.21 Your skills page

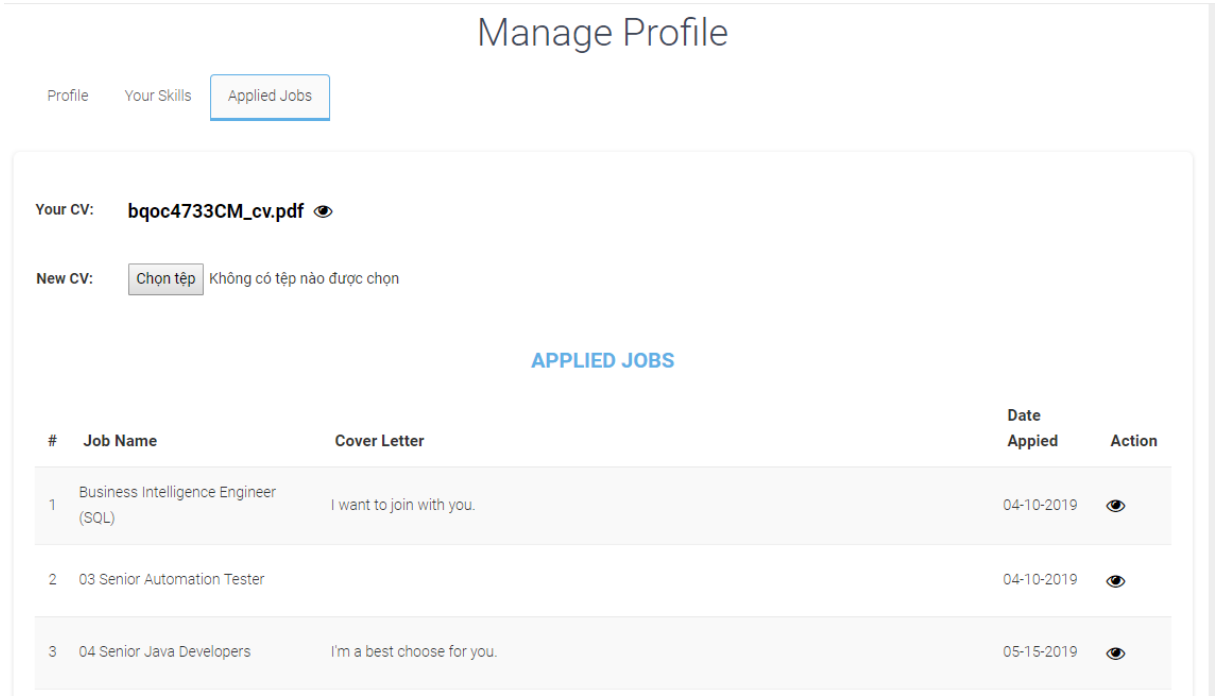

Figure 3.22 The applied jobs page

2) Enterprise

At the personal profile page, the enterprise can:

- Updating enterprise's profile
- Viewing the skill list
- Adding new skills
- Viewing the hiring job list
- Adding hiring posts

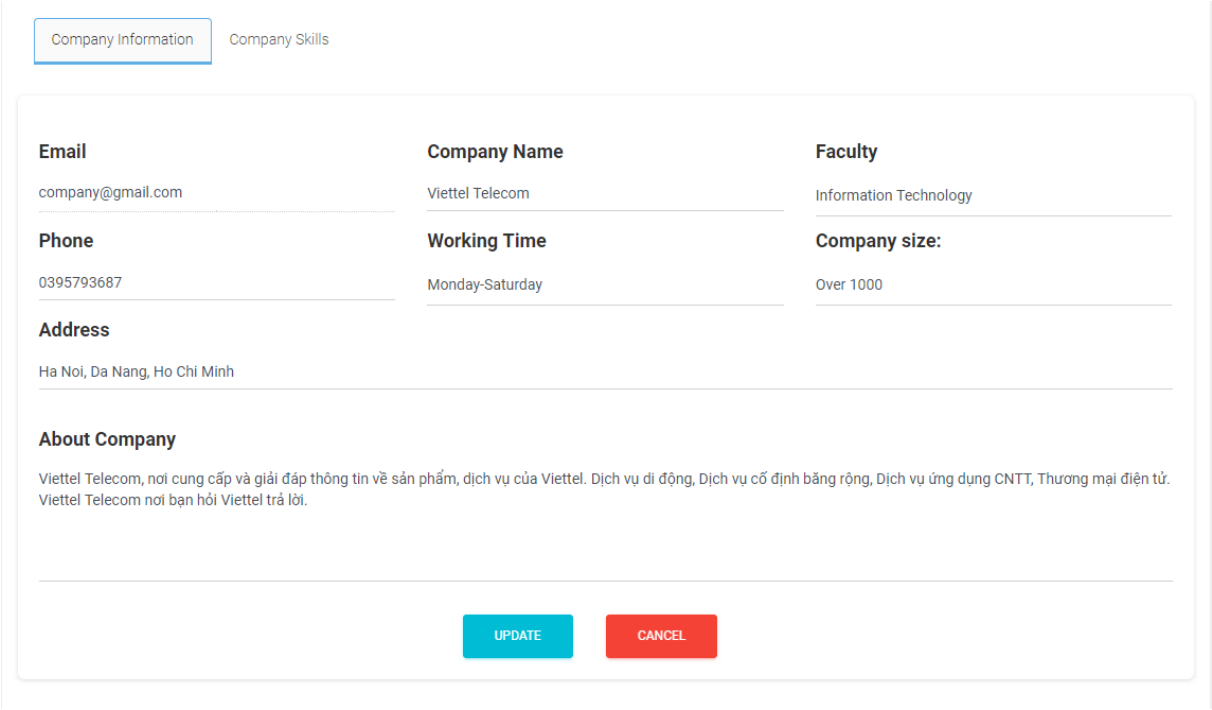

Figure 3.23 Enterprise profile page

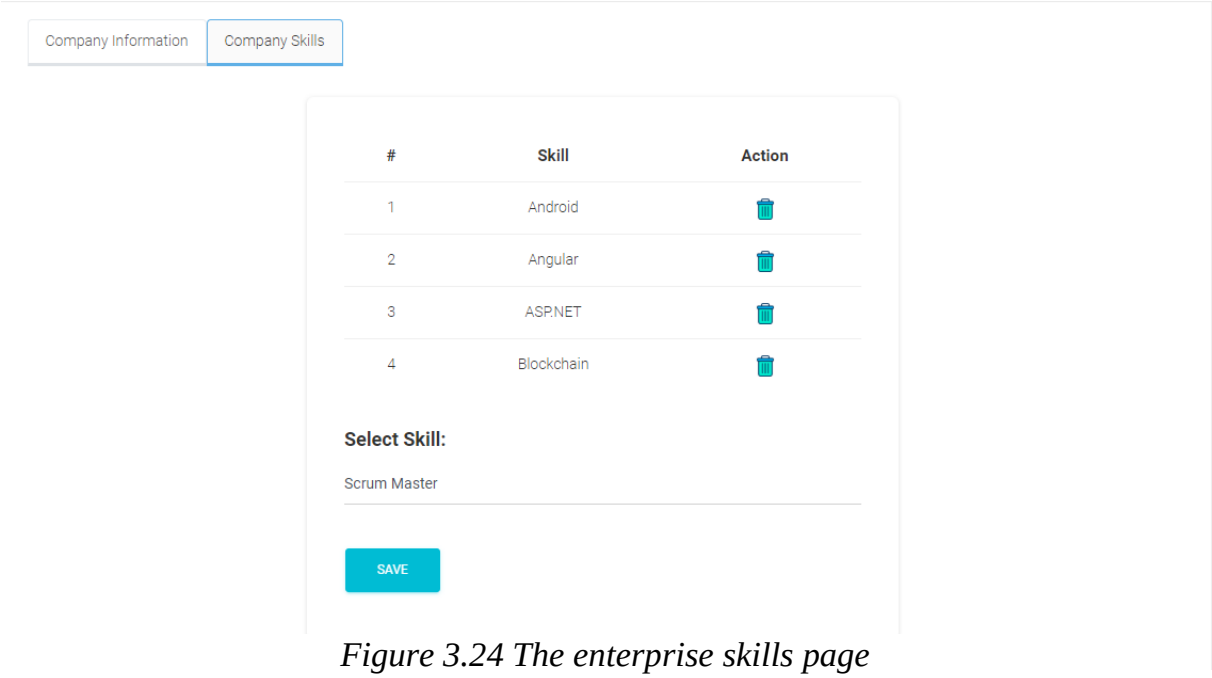

Figure 3.24 The enterprise skills page

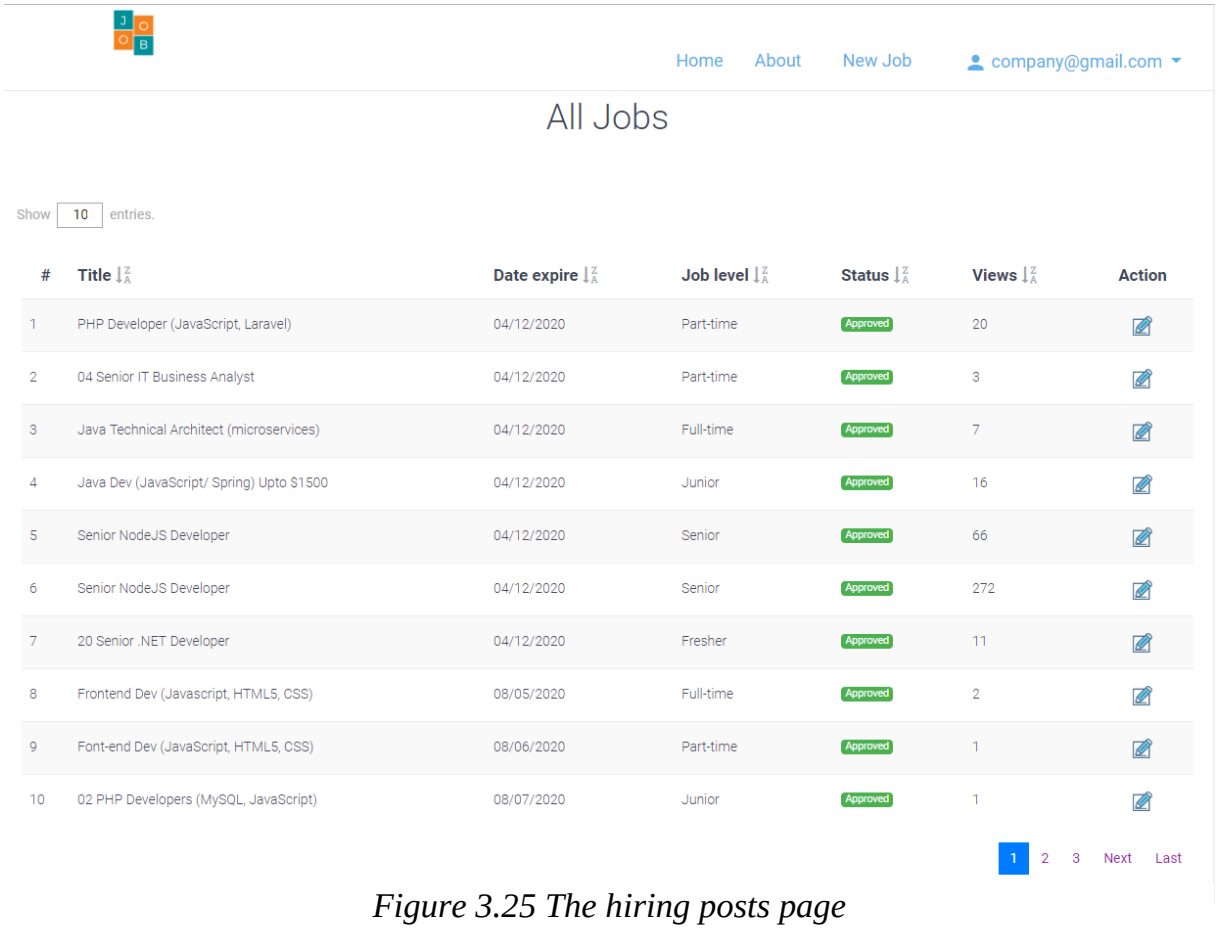

Figure 3.25 The hiring posts page

3.3.8. The admin system

After login successful, user is admin will be redirected to the dashboard page. At the dashboard page displays total student, total hiring post, total enterprise, and the admin list. From this page, admin manages all function of the system. At the left bar, admin clicks a path to redirect to another manage page.

With manage student function, the admin can view the student list, sort student list, view detailed a user, deactivate or activate any student (except an account has username is admin@gmail.com), etc.

With manage enterprise function, the admin can view the enterprise list, view detailed an enterprise, add a new enterprise, etc.

The admin also has to manage all hiring posts. A hiring post to be visible to the user, it must be accepted by the admin.

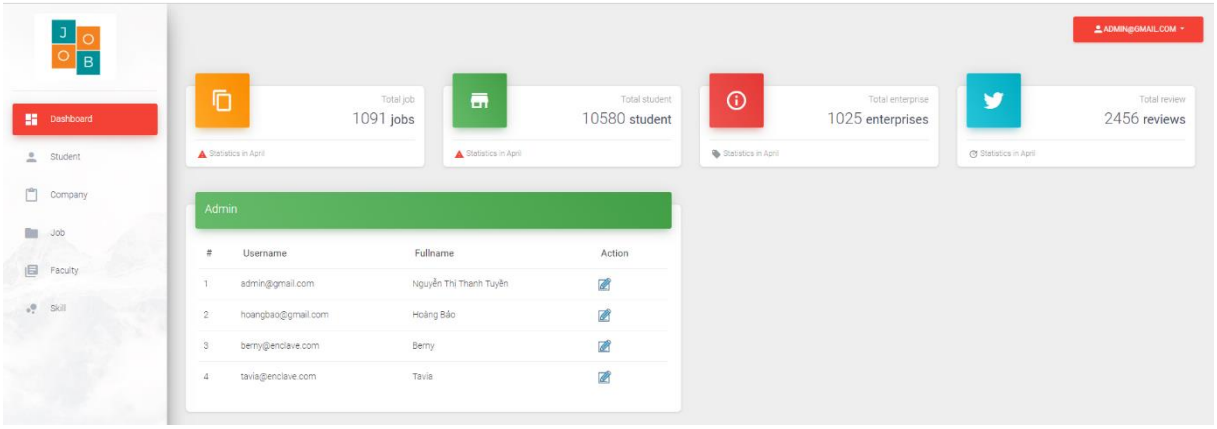

Figure 3.26 The dashboard page

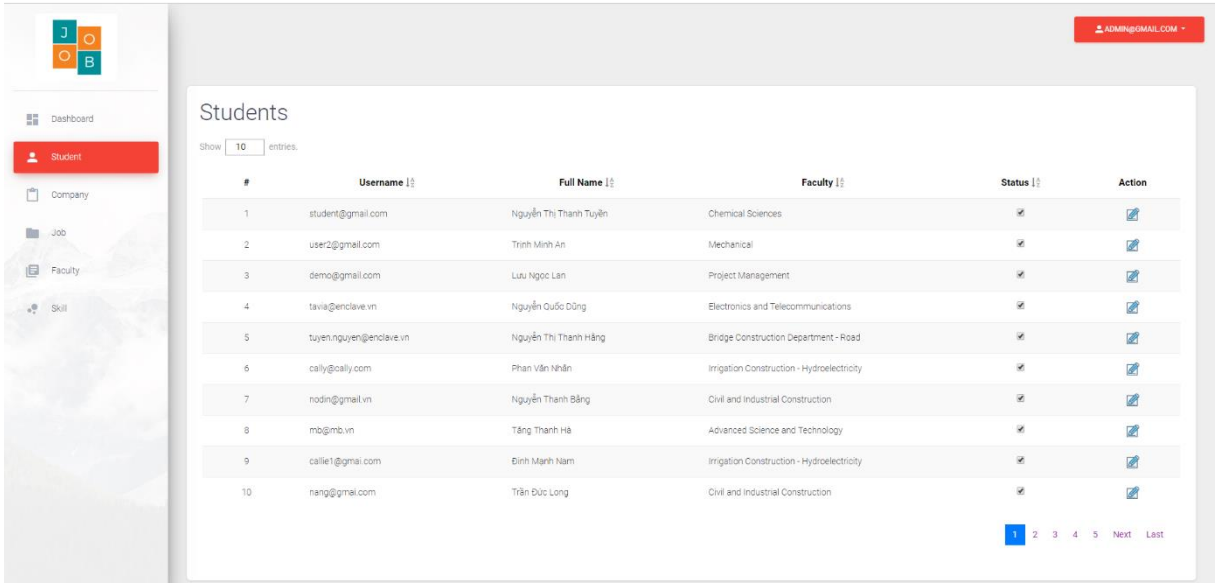

Figure 3.27 The student list page

3.4. System evaluation

3.4.1. Advantages

Basically, “DUT JOB - An application connecting DUT’s students and enterprises” meets the needs in the scope of the graduation project thesis.

In terms of UX/UI, my system provides simple and familiar UIs for users to interact and get information.

In this system, users can search and apply for a job online. They can search for information about enterprises and jobs offered by them. Enterprises can also create hiring posts and receive candidates' resumes via email.

Especially in my system, diversity of jobs is categorized by faculties (Information Technology, Electrical, Environment, etc.), by levels (internship, fresher, etc.), so it is easy for you to find. For students, find an internship position while being a student to gain experience is very necessary. Based on that need, administrators will focus on finding businesses that want to recruit interns to provide jobs for students.

3.4.2. Disadvantages

Besides the advantages mentioned above, my system still has some disadvantages:

- The synchronization of the system is not so stable and the performance is not too high.
- The system does not allow search by location.

CONCLUSION

1. Achievements

About the project, I have already followed the plan which was created in the first week of graduation project duration. I also try to make the system the most flexible whose purpose is to enhance user experience.

I made this system to helps students of DUT can find a job that fits their skills. The system also helps enterprises find a lot of candidates that suitable for their companies.

This system includes a lot of features:

- Registering a new account
- Searching for jobs by many criteria: faculty, level, job's name, skills
- Searching and viewing detailed information of enterprises
- Applying for a job online
- Viewing 4 faculties have the most hiring posts
- Viewing 8 latest hiring posts
- Viewing detailed a hiring post
- Managing personal profile for students: CV, applied job list and skills
- Managing system for enterprise: enterprise's information, and hiring posts
- Managing system for admin

After building this system, I could understand how to make a website application by Angular language. I also learn how to build a RESTful API with .NET Core 2.1 framework.

I also improved my skills a lot, including research skills, technical skills, presentation skills, English and many other soft skills.

2. Limitations

Although certain results have been achieved, there are still many challenges, risks, and issues with products:

- About search feature, users can't search by multiple keywords at once, one keyword only.

- Enterprises can’t manage candidates on the website, they only receive applications via email.
- The automatic feature sends an email for students (up to 1 mail per day) when having a lot of jobs which fit has not been implemented. The email just is sent at certain times. It makes students miss a lot of good jobs.

3. Future developments

With some disadvantage I have already mentioned in the evaluation part, I am aware of some development directions for this system in the future, such as:

- Developing application on mobile platforms
- Suggesting jobs that fit the skills of a student and recommend available candidates for job ads
- Building a network social for students of DUT
- Improving system processing speed, especially searching speed

REFERENCES

[1] https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.1

[2] https://www.c-sharpcorner.com/article/asp-net-core-2-architecture-design-pattern-ideology/

[3] https://angular.io/docs

[4] https://docs.microsoft.com/en-us/sql/sql-server

[5] http://www.sqlservertutorial.net/getting-started/what-is-sql-server/

[6] https://itviec.com

[7] https://www.vietnamworks.com/