

# **Recruitment BloQs: A Blockchain Academic Record Based Recruitment System**

By

Malvern Chatikobo

(H190305E)

HIT400 Capstone project Submitted in Partial Fulfillment of the

Requirements of the degree of

Bachelor of Technology

In

**Software Engineering**

In the

**School of Information Sciences and Technology**

Harare Institute of Technology

Zimbabwe



Ms. L Amos

.....  
05/2023

## HIT 400 Project Documentation Marking Guide

ITEM	TOTAL MARK /%	ACQUIRED/%
<b>PRESENTATION-</b> Format-Times Roman 12 for ordinary text, Main headings Times Roman 14, spacing 1.5. Chapters and sub-chapters, tables and diagrams should be numbered. Document should be in report form. Range of document pages. Between 50 and 100. Work should be clear and neat	5	
<b>Pre-Chapter Section</b> Abstract, Preface, Acknowledgements, Dedication & Declaration	5	
<b>Chapter One-Introduction</b> Background, Problem Statement, Objectives – smart, clearly measurable from your system. Always start with a TO... Hypothesis, Justification, Proposed Tools Feasibility study: Technical, Economic & Operational Project plan –Time plan, Gantt chart	10	
<b>Chapter Two-Literature Review</b> Introduction, Related work & Conclusion	10	
<b>Chapter Three –Analysis</b> Information Gathering Tools, Description of system. Data analysis –Using UML context diagrams, DFD of existing system Evaluation of Alternatives Systems, Functional Analysis of Proposed System-Functional and Non-functional Requirements, User Case Diagrams	15	
<b>Chapter Four –Design</b> Systems Diagrams –Using UML Context diagrams, DFD, Activity diagrams. Architectural Design-hardware, networking Database Design –ER diagrams, Normalized Databases Program Design-Class diagrams, Sequence diagrams, Package diagrams, Pseudo code Interface Design-Screenshots of user interface	20	
<b>Chapter Five-Implementation &amp; Testing</b> Pseudo code of major modules /Sample of real code can be written here. Software Testing-Unit, Module, Integration, System, Database & Acceptance	20	
<b>Chapter Six –Conclusions and Recommendations</b> Results and summary, Recommendations & Future Works	10	
<b>Bibliography –Proper numbering should be used.</b> Appendices –templates of data collection tools, user manual of the working system, sample code, research papers	5	
	100	/100

## Certificate of Declaration

This is to certify that work entitled “HIT400 Research Topic “is *submitted in partial fulfillment of the requirements for the award of Bachelor of Technology (Hons) in Software Engineering, Harare Institute of Technology. It is further certified that no part of research has been submitted to any university for the award of any other degree.*



(Supervisor)	Signature.....	Date.....
(Mentor)	Signature.....	Date.....
(Chairman)	Signature.....	Date.....

## Abstract

The education system in Zimbabwe has long been recognized for producing highly skilled and talented individuals. However, the problem lies in the limited opportunities for employment, especially in relation to internship programs, graduate trainee positions, and full-time employment. It has been observed that nepotism and bribery play a significant role in the recruitment process, thereby leading to qualified candidates being overlooked in favour of those with connections or those who offer bribes.

To address this issue, this project proposes the need for a blockchain-based recruitment platform that can provide a fair and transparent hiring process. The platform will be web-based and will allow companies to search for viable candidates based on their university-verified academic results. The proposed system will leverage blockchain technology, which provides a decentralized and tamper-proof system, ensuring the validity and authenticity of candidate data.

This project argues that the adoption of blockchain technology in the recruitment process can help to restore trust and credibility in the system, providing a level playing field for all candidates. Moreover, the use of blockchain technology can facilitate faster and more efficient recruitment processes, leading to reduced costs and increased productivity. Ultimately, this project aims to contribute to the ongoing discourse on the potential of blockchain technology to revolutionize the recruitment process.

## Preface

It gives me immense pleasure to present this project, which is the culmination of months of research, hard work, and dedication. The project explores the need for a recruitment platform based on blockchain technology to tackle the issue of nepotism and bribery in the recruitment process, which is prevalent in Zimbabwe.

The project was born out of a passion for creating a positive impact in the education and employment sectors in Zimbabwe. As a student in the software department, I witnessed firsthand the struggle that highly skilled individuals face in securing employment opportunities due to the unfair recruitment practices prevalent in the workplace.

This project is an attempt to address this problem by proposing a blockchain-based recruitment platform that will enable companies to find qualified candidates based on their university-verified academic results. The platform aims to create a fair and transparent recruitment process that eliminates the need for bribery and nepotism, thereby giving every candidate an equal opportunity to showcase their skills and secure employment.

## Acknowledgements

First and foremost, I would like to thank God, whose grace has been sufficient from that first day of pre-school to this milestone I'm on the verge of achieving.

I am thankful to my supervisor, Ms. L Amos, for her invaluable guidance, expert advice, and unwavering support throughout the project. Her mentorship has been instrumental in shaping the direction of this work and has helped me to develop important skills and knowledge in the field.

Furthermore, I would like to express my gratitude to the staff of the Software Engineering department for their assistance and support, especially during the challenging times of the project. The department's resources and facilities have been crucial in carrying out the research and development tasks.

I would like to thank my mother and aunt for being warriors and fighting for me and my siblings to have a quality education. I would also like to thank the rest of my family and friends for their unwavering support, encouragement, and love. Their constant belief in my abilities has been a driving force in overcoming challenges and achieving success.

Finally, I would like to thank all the participants who volunteered their time and efforts to provide valuable data for this project. Their cooperation and contribution have been instrumental in the successful completion of this work.

Once again, I extend my sincere appreciation to everyone who has played a role in this project. Your support and contributions have been invaluable, and I am deeply grateful for your presence in my life.

## Dedication

I dedicate this project to family, my family. You guys have been there since day one. I've learnt, laughed and loved with you. It's been a wonderful 23 years with you. Thank you for all the values you've taught me.

I would like to take this time to mention my mom and aunt, you've been amazing parents, I wouldn't be here if it wasn't for you. Thank you for protecting me from the things I didn't know I needed protection from and putting me onto a path that's led me to this point. I appreciate everything you do for us.

I would like to thank my brothers, for standing in where mom couldn't. Thank you for erasing family lines, and being united as you are. Thank you for being an inspiration to us all. Congratulations to the newlyweds, Mr. & Mrs. Chatikobo, and Mr. Man kanya, we're waiting in anticipation for the big day.

I'd like to give a shout out to my friends, and connections I made for the past 4 years.

## Table of Contents

Certificate of Declaration.....	ii
Abstract .....	iii
Preface .....	iv
Acknowledgements .....	v
List Of Tables .....	xii
List Of Figures .....	xiii
1 Chapter One-Introduction.....	1
1.1.1 Background.....	1
1.2 Problem statement.....	1
1.3 Aim.....	2
1.4 Objectives.....	2
1.5 Justification .....	2
1.6 Proposed Tools.....	2
1.7 Feasibility study .....	3
1.7.1 Technical.....	3
1.7.2 Economic .....	3
1.7.3 Operational.....	3
1.8 Project plan.....	4
1.8.1 Time plan .....	4
1.8.2 Gantt chart.....	4
2 Chapter Two-Literature Review .....	5
2.1 Abstract .....	5
2.2 Introduction .....	5



2.3	Literature Review .....	6
2.4	Recruitment and Selection .....	6
2.5	Blockchain Systems .....	6
2.6	Why there is need of a Recruitment platform? .....	7
2.7	Related work .....	7
2.8	Solution .....	11
3	Chapter Three –Analysis .....	13
3.1	Information Gathering Tools.....	13
3.1.1	Surveys.....	13
3.1.2	Going through documentation .....	13
3.2	Description of system.....	13
3.3	Data analysis .....	14
3.3.1	Context diagram.....	14
3.3.2	DFD of existing system .....	15
3.4	Evaluation of Alternatives Systems .....	15
3.4.1	EKO Service by echolink.....	15
3.4.2	BambooHR .....	15
3.4.3	TestGorilla .....	16
3.4.4	QualiChain .....	16
3.5	Functional Analysis of Proposed System.....	17
3.5.1	Functional Requirements .....	17
3.5.2	Non-functional Requirements .....	17
3.6	User Case Diagram for Proposed System .....	19
4	Chapter Four –Design.....	20
4.1	Systems Diagrams .....	20

4.1.1	Context diagrams .....	20
4.1.2	DFD.....	21
4.1.3	Activity diagrams.....	22
4.2	Architectural Design .....	23
4.2.1	Hardware.....	23
4.2.2	Networking .....	24
4.3	Database Design.....	24
4.3.1	Normalized Databases .....	24
4.3.2	ER diagrams.....	27
4.4	Program Design.....	28
4.4.1	Class diagrams .....	28
4.4.2	Sequence diagrams.....	30
4.4.3	Package diagrams.....	31
4.4.4	Pseudo code .....	31
4.5	Interface Design-Screenshots of user interface .....	32
5	Chapter Five-Implementation & Testing.....	35
5.1	Software Testing- .....	35
5.1.1	Unit Testing .....	35
5.1.2	Module Testing .....	37
5.1.3	Integration Testing .....	39
5.1.4	System Testing.....	39
5.1.5	Database Testing.....	40
5.1.6	Acceptance Testing.....	40
5.2	Sample Code .....	40
5.2.1	Universities Smart Contract.....	40

5.2.2	Search and Ranking Functions.....	41
6	Chapter Six –Conclusions and Recommendations .....	47
6.1	Results and summary .....	47
6.2	Recommendations .....	47
6.3	Future Works.....	47
7	Appendices .....	48
7.1	References .....	48
7.2	Templates of data collection tools.....	50
7.3	User manual of the working system.....	51
7.4	All users.....	51
7.4.1	Log in.....	51
7.5	Universities.....	51
7.5.1	Adding a university.....	51
7.5.2	Adding a Company .....	51
7.5.3	Adding A Course .....	51
7.5.4	Uploading Information.....	51
7.6	Company .....	52
7.6.1	Creating a job opening .....	52
7.6.2	Adding job candidates.....	52
7.6.3	Sending out interview invitations .....	52
7.6.4	Add employee review. ....	52
7.7	Sample code .....	53
8	Technical paper.....	57



## List Of Tables

Table 1:Time Plan .....	4
Table 2:Gantt Chart.....	4
Table 3: Companies Table .....	24
Table 4:Job opening candidate choices table.....	24
Table 5: job openings table .....	25
Table 6: Personal skills .....	25
Table 7: universities .....	25
Table 8: university courses .....	26
Table 9:OBJECTIVE ACHIEVED.....	60

## List Of Figures

Figure 1: Use case definition of the Fenix system. ....	8
Figure 2: Use case definition .....	8
Figure 3: data movement outside the BOLL system. ....	9
Figure 4: Proposed design of blockchain of learning logs (BOLL).....	10
Figure 5: Echo System Overview from [13].....	11
Figure 6: Context Diagram .....	14
Figure 7: Data flow Diagram of existing system .....	15
Figure 8: 3.6 User Case Diagram for Proposed System .....	19
Figure 9: Context Diagram of Proposed Sytem.....	20
Figure 10: Data Flow Diagram for Proposed System .....	21
Figure 11: Activity Diagram.....	23
Figure 12: ERD .....	28
Figure 13: Class Diagrams.....	28
Figure 14: Sequence Diagram.....	30
Figure 15: Package Diagram.....	31
Figure 16: Job Candidates Page.....	32
Figure 17: Candidate Profile 1 .....	32
Figure 18: Candidate Profile 2.....	33
Figure 19: Add Course Page .....	33
Figure 20: Upload Students Page.....	34
Figure 21: Upload Module Resultsts .....	34
Figure 22: Survey Questions.....	50
Figure 23: Use case definition of the Fenix .....	58
Figure 24: Use case definition of the QualiChain system.....	58
Figure 25: This shows an example of data movement outside the BOLL system.....	59
Figure 26: Proposed design of blockchain of learning logs (BOLL).....	59
Figure 27: The interface of an individual's profile .....	60

# 1 Chapter One-Introduction

## 1.1.1 Background

Recruitment practices in most of the business world in Zimbabwe are not systematic and less than fair. This leads to many undeserving and unqualified people getting employed rather than the right people for the job. Since there is no platform for job seekers to go and see all the jobs they qualify for. This means finding vacancies open is basically random and by luck. This to some extent contributes to unemployment. Job adverts in Zimbabwe are still predominantly paper based. This is ineffective since most of the country has gone digital and thus paper-based ads reach a smaller audience.

The application process for jobs nowadays involves sending emails to the hiring organization. This means the HR department of the hiring company has to constantly check their email for any new applications. It therefore becomes inevitable that some emails might be missed if careful attention is not put in checking the emails.

After all the applications have been pooled together, the HR department of the organization has to go through all the applications and the accompanying documents to create a short list of the applicants that meet their academic and experience requirements for the job vacancy. They then have to contact the selected applicants back to inform them of any further details pertaining the job they applied for and the following steps in the recruitment process.

## 1.2 Problem statement

The recruitment process in Zimbabwe is riddled with favoritism and non-systematic practices. This, to some extent, contributes to unemployment and consequently brain drain. With the recruitment current process, if a job gets high volumes of applicants, it will take a long time to create a short list of candidates who qualify who will be invited for interview.

Problems currently faced during the recruitment process include:

- Faked academic qualifications
- Companies poor advertising of job openings
- Time consumed in analyzing resumes and verifying academic records and job histories

- most applicants that apply for jobs they are not directly qualified for among others.
- Lack of a pool of experts to choose from
- Startups do not get as many applicants to be sure they have the exact qualities the need

### 1.3 Aim

To develop a blockchain based academic record platform for selection in recruitment for use by companies to find employment eligible university students (current or past)

### 1.4 Objectives

- To develop a blockchain system to store relevant tertiary institute academic records and employment history records.
- To develop a system which can be used to verify any information pertaining skills and employment history of an individual.
- To develop a web platform for companies to find employment eligible university students (current or past)
- To develop a system that can map a student's proficient areas to an area of specialization for easier display to influence the decision making.

### 1.5 Justification

The system to be developed provides a number of advantages in the recruitment process. It will have analysis tools which will be used to compare viable employee candidates that will help make the best employment choice. Since the productivity of an organization is directly linked to its workforce, hiring the right person is important. The solution will help organizations find the right person. The system will promote transparency since data cannot be altered thus no one can temper with it. This will mean any information on the system is to be considered true and verifiable.

### 1.6 Proposed Tools

- ❖ Solidity
- ❖ Remix IDE/Truffle
- ❖ Web3.py
- ❖ Python Flask



## ❖ MetaMask

### 1.7 Feasibility study

#### 1.7.1 Technical

With the emergence of blockchain technology, a lot of frameworks have been developed for developers to use in developing Dapps. This means a lot of support and infrastructure is available for the development of blockchain applications. The developer has a consolidated understanding of blockchain technology and how blockchain networks operate and store data. This will be paramount in the development of the proposed system. The developer also has a strong understanding of web development technologies which will be paramount in the development of the platform's interface.

#### 1.7.2 Economic

The project is economically feasible, given the fact that the requirements for the project have already been met by tools the developer already has access to. The project if implemented has an opportunity to be monetized on a subscription basis since the product will provide a service that will bring productivity to organizations and stakeholders.

#### 1.7.3 Operational

After extensive research, the solution showed multiple advantages in its implementation in the recruitment process of companies in Zimbabwe. The use of a blockchain ledger makes the falsification of any information impossible and will be paramount in holding hiring parties accountable when favoritism is suspected. The solution is paramount since it can be used create a standard for recruitment in the country.

1.8 Project plan

1.8.1 Time plan

Table 1:Time Plan


1.8.2 Gantt chart

Table 2:Gantt Chart

Month	August	September	October	November	January	February	March	April	May
January									
February (2020)									
February (2021)									
March									
March to October									
October									
November									
December									
January (2021)									

## 2 Chapter Two-Literature Review

The need for a Blockchain based academic record system used in recruitment.

### 2.1 Abstract

The education system in Zimbabwe produces highly skilled individuals who don't get the same employment opportunities. This applies to both student recruitment for the internship programs and graduates for graduate trainee positions or full-time employment. Nepotism and bribery usually control the recruitment processes in the workplace. This literature review will assess the need of a recruitment platform. This paper will show why the platform needs to be based on blockchain technology. The proposed system is a web-based platform that will allow companies to find viable candidates to the position they need filled based on their university verified academic results.

**Key Words:** Blockchain, recruitment platform, internship

### 2.2 Introduction

According to [1] Recruitment is the process to fully fill the offered work positions in sufficient number and qualities of the applicants, as well as to meet the expectations and requirements of the organization. Recruitment is using the analysis of the job in order to select and identify the needs and requirements of the organization. Before the internet was a prominent part of people's lives, recruitment used to be done by means of paper-based ads in newspapers, magazines, or via radio or tv ads, and in cases of students or recent graduates through universities. Recruitment has however become more ICT centric and continue to be more and more so. This is because organizations find it to be more effective, faster and less costly. Organizations therefore use online to adverts to lure candidates [2]. The use of an online platform for according to [3] selection and recruitment ensures accuracy of information. This of course is referring to the traditional recruitment platforms, and the information being referred to is that provided by applicant/ job seeker and is usually not verifiable.

The proposed system will be based on blockchain technology. Blockchain technology is defined as an immutable distributed public used to store data over nodes in a peer-to-peer network [3]. Data/transactions stored on a blockchain is tamper proof and secure [4]. Blockchains provide a decentralized way of storing data with persistency (ability to check the authenticity of a chain),

anonymity (ability to conceal data that should not be public) and auditability (can track when a transaction was made) [4].

## 2.3 Literature Review

A company's main asset is its work force. It is therefore paramount that companies find the right candidate, with the right qualifications for a job. Having the right staff leads to productivity, which in turn leads to profitability [5].

## 2.4 Recruitment and Selection

An organization performs a job analysis to determine the skills they need for a job. They then analyze the amount of manpower needed so that they have the correct amounts of employees of a certain skill set. From the advertising, applications, application analysis and interviewing then takes place [6]. The current recruitment process involves the company looking to hire workers sending out job adverts. Applicants usually therefore send their CVs (Curriculum Vitae) to the company in line with the job advert. These CVs usually contain a person's personal details, academic results qualifications and skills that maybe both technical and non-technical. The company pools together those results applicants together and analyze all the CVs to create a short list of applicants that look to have the right qualifications they need. This process is greatly inefficient and time consuming when the number of applicants is high. This is because the CVs might not follow the same format and therefore it takes time analyze them all. This applies to both interns and full-time employees [1].

In other cases, this process becomes affected by nepotism and sometimes bribery. This means that more deserving candidates don't get afforded the opportunity to work at higher profile companies.

## 2.5 Blockchain Systems

A block chain is a distributed peer-to-peer ledger that contains records called blocks of data linked by an encrypted hash i.e., a block contains the hash of the previous block. Blockchain technology was first introduced in 2008 where it was used to create a digital currency (bitcoin) [3]. From then it has evolved and has seen the emergence of many other digital currency. Blockchain technology has however found more applications in different other sectors including

cryptocurrency, Internet of Things (IoT), healthcare, cyber security, insurance, asset tracking among other things [4].

This technology has seen much more interest in the recent years due to its multiple desirable qualities. These qualities include high speed of validating and verifying data, high security, transparency all thanks to the available tamper proof mechanisms built into the network [7]. This makes it suitable to develop systems where data added to it needs to remain unchanged thus proving its validity.

## 2.6 Why is there need of a Recruitment platform?

ICT is adopted in many sectors to improve the efficiency thereof. The Human resource industry has adopted the use of IT in its many functions from Recruiting, hiring, and onboarding, Human resource planning, Benefits and compensation, Performance management to Training and development. Its application in recruitment has considerable from reducing time taken to apply for a job to enabling smooth communication between the applicants and the company [8]. The use of ICT has seen platforms where companies can advertise job openings while there are others where professionals can have profiles that show their qualifications and employment history [9]. These platforms however still lack in efficiency, transparency and in times ways to verify the information.

According to [10] there are problems faced by companies in hiring new staff. These problems include a company's poor branding making it less attractive to applicants, fake resumes, lack of verified information, most applicants that apply for jobs they are not directly qualified for among others.

Chhetri in [10] continues to add that a recruitment platform based on blockchain technology would reduce selection bias, prevent candidate(applicant) fraud, provide easy and faster access to verified information, provide access to a wider talent supply and decrease the cost of hiring new employees.

## 2.7 Related work

QualiChain as described in [11] is a Portuguese blockchain based application that is used to store diplomas issued by Higher Education Institutions (HEIs). It is used by to combat counterfeiting and falsification. It provides an interface for hiring companies to check the authenticity of an

applicant's credentials. It is an extension of an already existing system called Fenix which is an academic management platform that supports full program and course management, including the issuing of diplomas (essentially PDF documents that are printed, signed, and officially stamped).

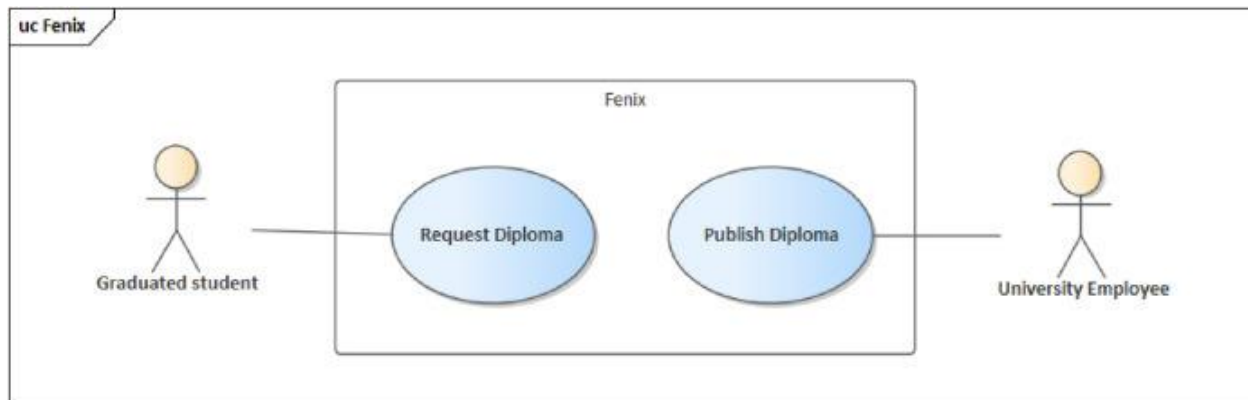


Figure 1: Use case definition of the Fenix system.

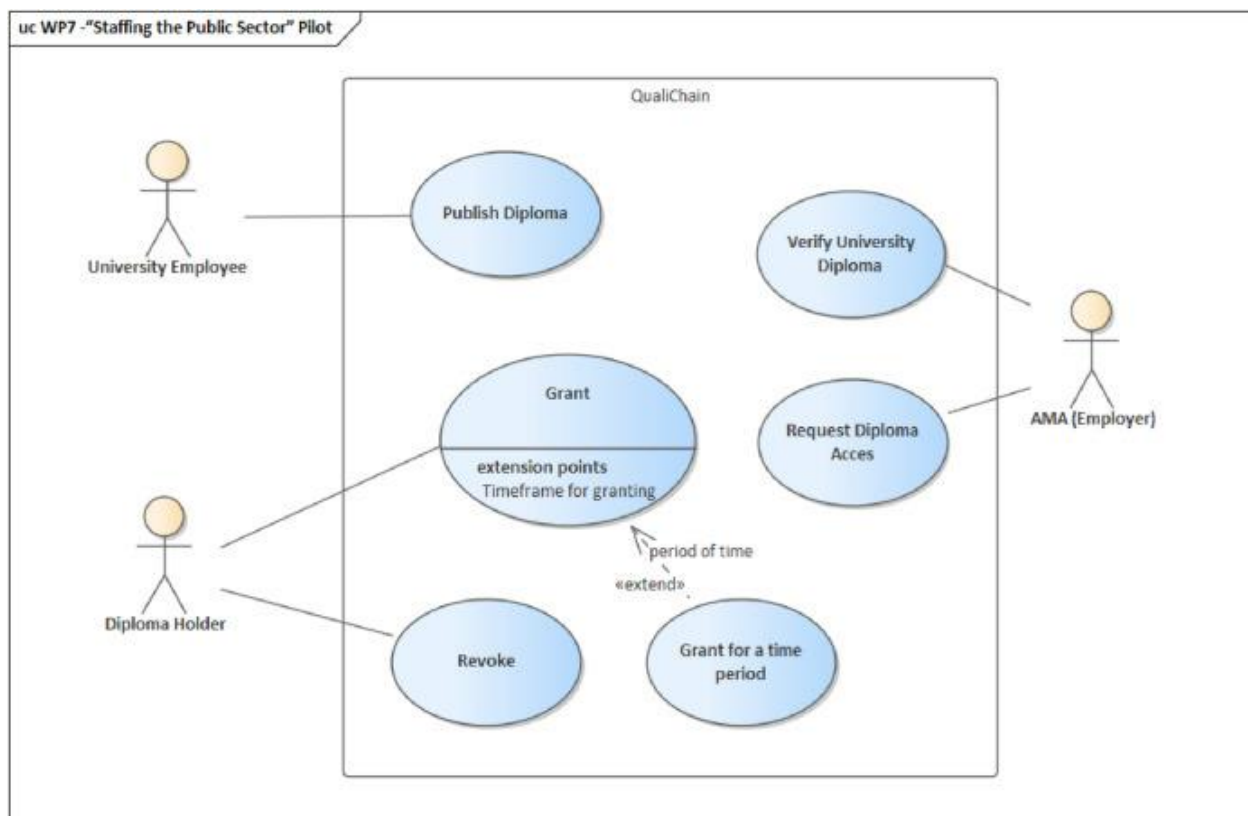


Figure 2: Use case definition

Use case definition of the QualiChain system involving the University employee, the Diploma holder, and AMA. The purpose is to specify the use cases of requesting, verifying, and revoking the diploma.

The blockchain of learning logs (BOLL) as proposed in [12] is a system that connects different institutions of learning and is used to transfer a student's records as they move between institutions. The system is used to keep track of a person's academic results and institutions attended in detail. It also solves the cold-start problem which is described as having a new academic record being created without being connected to the students' previous records.

The following 2 figures show how data is transferred between institutions outside the BOLL and on the BOLL respectively.

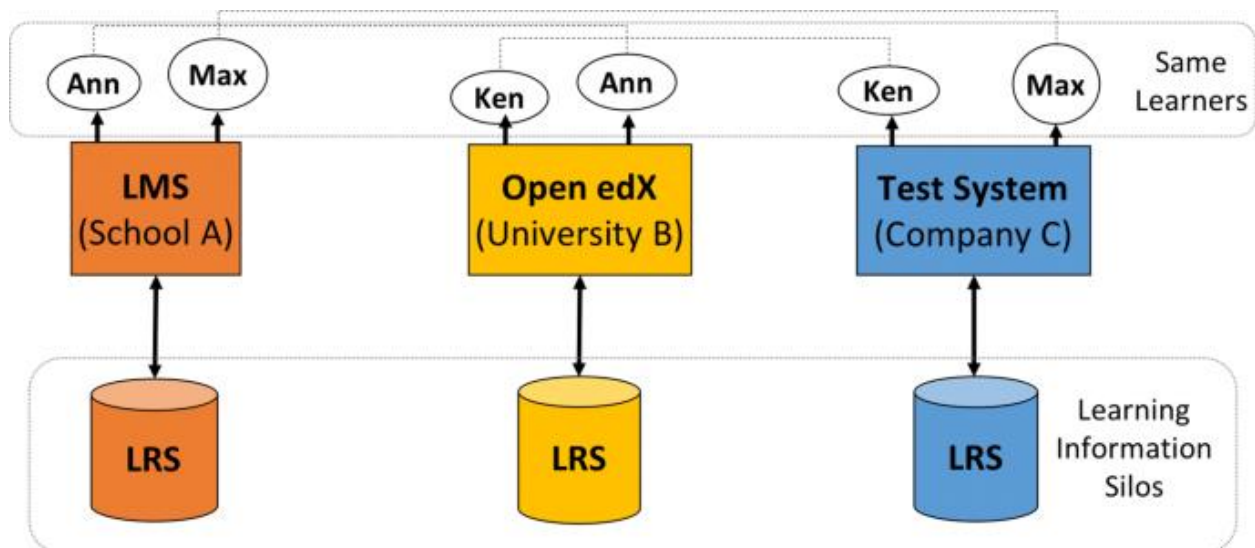


Figure 3: data movement outside the BOLL system.

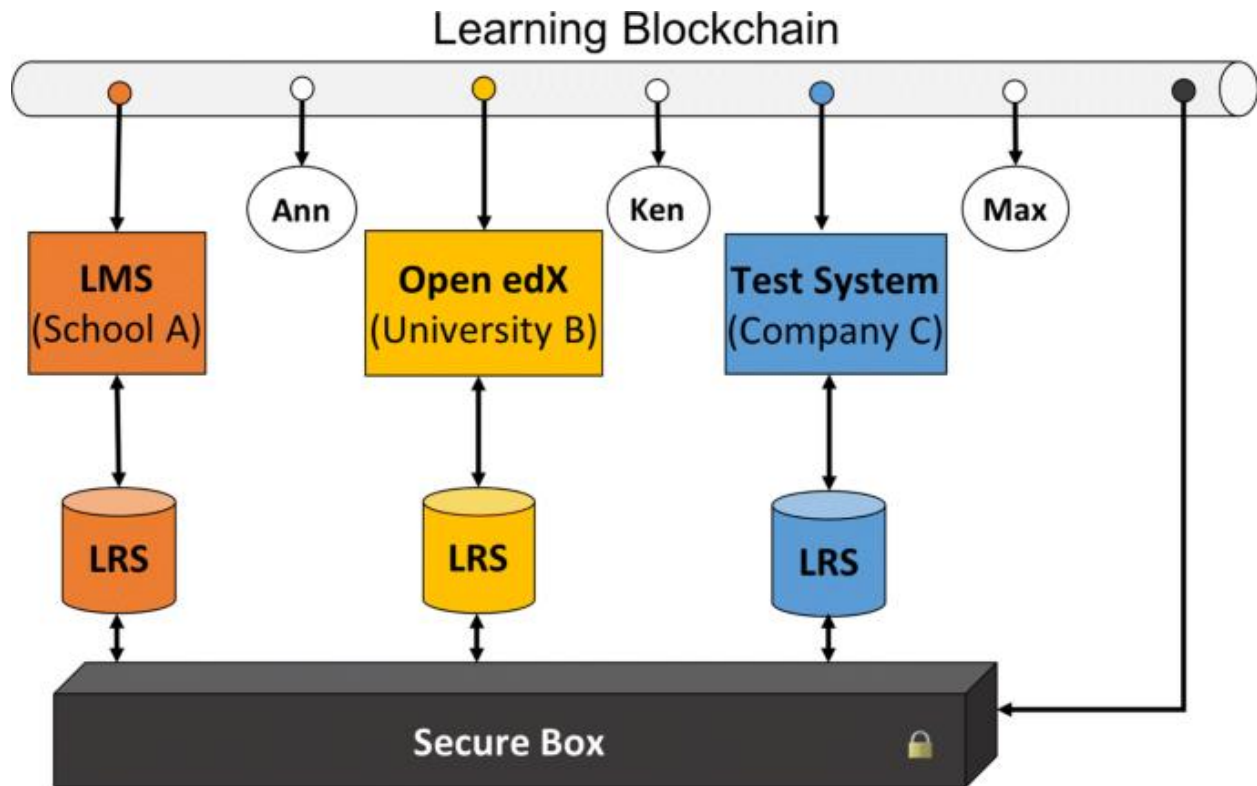


Figure 4: Proposed design of blockchain of learning logs (BOLL)

The EchoLink Service is a blockchain based system that provides a platform to verify education, skill, and work experience information. It makes use of blockchain technology's properties like permanence and time stamp functionality, EchoLink provides users with reliable information concerning the academic records and consequently skills, and work history of job candidates. the figure below shows the architecture of the EchoLink service.



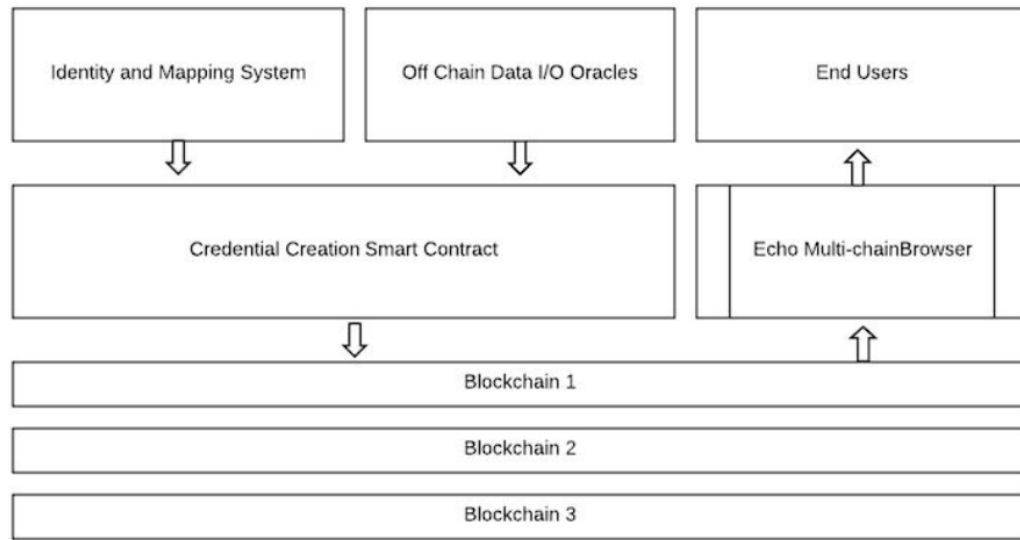


Figure 5: Echo System Overview from [13]

In [14] the writers examine the potential uses of blockchain technology. They suggest a couple of uses one of which was the use of Blockchain technology in storing academic records. These records could be shared by the owners in the form of a link to anyone they want to have access to the information. [15] goes on to add that this proposed system was first applied in the University of Nicosia. The paper states that the university issues academic certificates whose authenticity can be verified through the Bitcoin blockchain.

## 2.8 Solution

This therefore has prompted me to design a platform that will overcome these weaknesses. The platform will focus on tertiary education as the basis of any form of formal employment. The system will be based on blockchain technology which will make any information on it verifiable and valid. The system will contain academic results of students that are eligible to be employed as interns or that have graduated in the different fields. This platform is to be used to make the recruitment process easy and effective. With a pool of professionals to choose from, companies get to see only students or graduates with the qualifications they need for the job opening.

Companies will have to register with any of the tertiary institutes and when registered they can browse through viable candidates in the field of the position they want to fill. When browsing, companies initially have only access to a student's academic results and thus the skills they possess. Students will however through the use of smart contracts be able to determine what the

recruiters get to see. This platform aims to bring equality to the recruitment process and also reduce the amount of academic fraud.

## 3 Chapter Three –Analysis

### 3.1 Information Gathering Tools

#### 3.1.1 Surveys

Surveys are a standardized set of questions distributed to stakeholders to gather feedback often conducted online. They are useful when developers need to gather feedback from stakeholders who are geographically dispersed. Surveys can also be useful in gathering quantitative data and can be easier to analyze than qualitative data.

#### 3.1.2 Going through documentation

Going through documentation is another information gathering tool used in software development. It involves reviewing existing documents, such as user manuals, business requirements, or technical specifications. This tool is useful when developers need to understand how the system is currently being used or how it is supposed to be used. Reviewing documentation can also help identify potential conflicts or issues with existing requirements. One advantage of going through documentation is that it is often a quick and inexpensive way of gathering information. Developers can access existing documents without having to spend time or resources on creating new ones.

### 3.2 Description of system

The proposed system is a blockchain-based platform designed to address weaknesses in the recruitment process for tertiary education students and graduates. It aims to make the recruitment process easy and effective by providing companies with a pool of professionals to choose from, based on their qualifications and academic results.

The system will contain academic results of students eligible to be employed as interns or those who have graduated in different fields. These records will be stored on the blockchain, making any information on it verifiable and valid. The platform will bring equality to the recruitment process and reduce the amount of academic fraud.

To use the platform, companies must register with any of the tertiary institutes and browse through viable candidates in the field of the position they want to fill. Initially, companies will

have only access to a student's academic results and the skills they possess. However, through the use of smart contracts, students will be able to determine what the recruiters get to see.

Overall, the blockchain-based platform will provide a reliable and secure way to store and verify the academic records of students and graduates. It will make the recruitment process easier and more effective for companies and provide a fair and equal opportunity for students and graduates to showcase their skills and qualifications.

### 3.3 Data analysis

#### 3.3.1 Context diagram

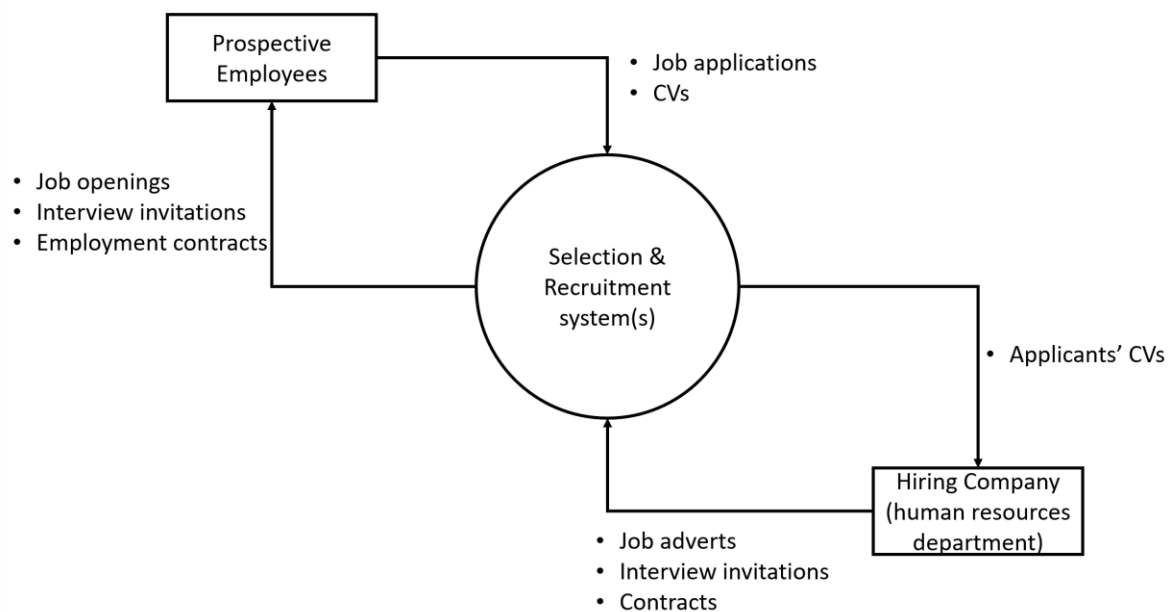


Figure 6: Context Diagram

### 3.3.2 DFD of existing system

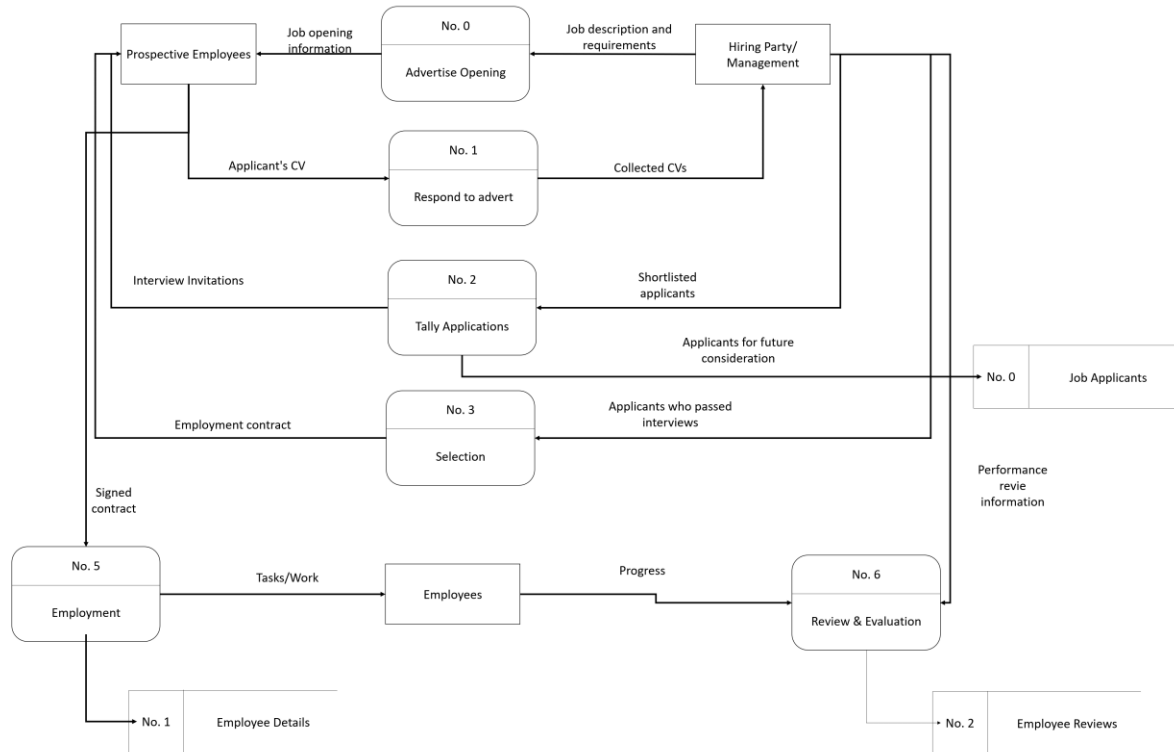


Figure 7: Data flow Diagram of existing system

## 3.4 Evaluation of Alternatives Systems

### 3.4.1 EKO Service by echolink

EKO is a blockchain based system that provides verified education, skill, and work experience information. Taking advantage of blockchain technology's immutability and time stamp functionality, EKO provides users with trusted information regarding a job candidate's education, skill, and work experience. EKO provides savings to recruiters in time and financial resources. The EKO system can also be used for additional industries, such as banking, finance, general notary service. EKO is building high-trust economy through innovations in blockchain technology.

### 3.4.2 BambooHR

BambooHR offers small and growing companies a human resource information system (HRIS) that includes an applicant tracking system (ATS), time tracking, payroll processing, employee

engagement and employee satisfaction tools, automated reminders, and workforce data analytics to manage all aspects of the employee lifecycle.

With BambooHR, any HR team or HR department can improve the hiring process, onboard new employees, manage compensation, maintain and analyze employee data, and develop company culture. Using an integrated ATS and onboarding tools, BambooHR streamlines the whole recruiting process from applications to first-day orientation.

It also provides customization services, multiple administrator roles, user support, and inactive employee management. BambooHR pricing uses a subscription-based model that depends on the number of employees plus a one-time implementation fee.

#### 3.4.3 TestGorilla

TestGorilla is a human resource (HR) management software designed to help businesses improve hiring processes by conducting pre-employment tests for potential candidates. Professionals can create personalized assessments by adding the applicant's name and job role and automatically receive recommendations about relevant tests.

Using TestGorilla, managers can add long-form or multiple-choice questions in assessments and send invitation links to multiple candidates via email. Administrators can track the real-time progress of applicants and sort them based on ratings, alphabetical order, latest activity or custom responses. Additionally, it lets users review candidates' test scores, add private notes and generate detailed performance reports.

#### 3.4.4 QualiChain

QualiChain is a blockchain based project that focuses on storage, of tertiary education information, work history and provides features for selection in the recruitment process. It is used for staffing purposes. QualiChain provides personalized candidate notifications for job openings by matching individual profiles with available jobs. It also improves the efficiency of the recruitment process, through fast and secure validation of the academic and professional qualifications of job candidates. It provides decision support functionality is used to improve the credibility and flexibility of the ranking of job candidates. QualiChain supports the planning, implementation, and coordination of human resources management policies. It speeds-up the

recruitment process by providing validated CV information and facilitates the selection and mobility process using semantic matching methods. It also helps in the identification of skills gaps in employees and suggests training for skills development.

### 3.5 Functional Analysis of Proposed System

#### 3.5.1 Functional Requirements

- The system must allow tertiary institutes to upload relevant academic records (courses, registration, results) from one file exported from their database system (.sql or .csv files)
- The system must allow registered companies to search for employment eligible individuals.
- The system should sort candidates by their academic results and also their rating by previous employers.
- The system should show an individual's area of specialization based on their academic results.
- The system should allow students to share links or QR code that allow the holder to view and verify their academic records and employment history.

#### 3.5.2 Non-functional Requirements

- Users can use tertiary based login or can request independent login credentials with an initial system generated password. Users must change the initially assigned login password immediately after the first successful login. Moreover, the initial should never be reused.
- Users must not be able to update any academic or work history information.
- The system must allow multiple accounts per organization for collaboration purposes.
- The software should be portable to allow users to continue selection processes on any device even when away from work.
- The system should allow fast access of all recruitment relevant information.

- Personal information about users should be obstructed until a decision to select the individual has been made.



### 3.6 User Case Diagram for Proposed System

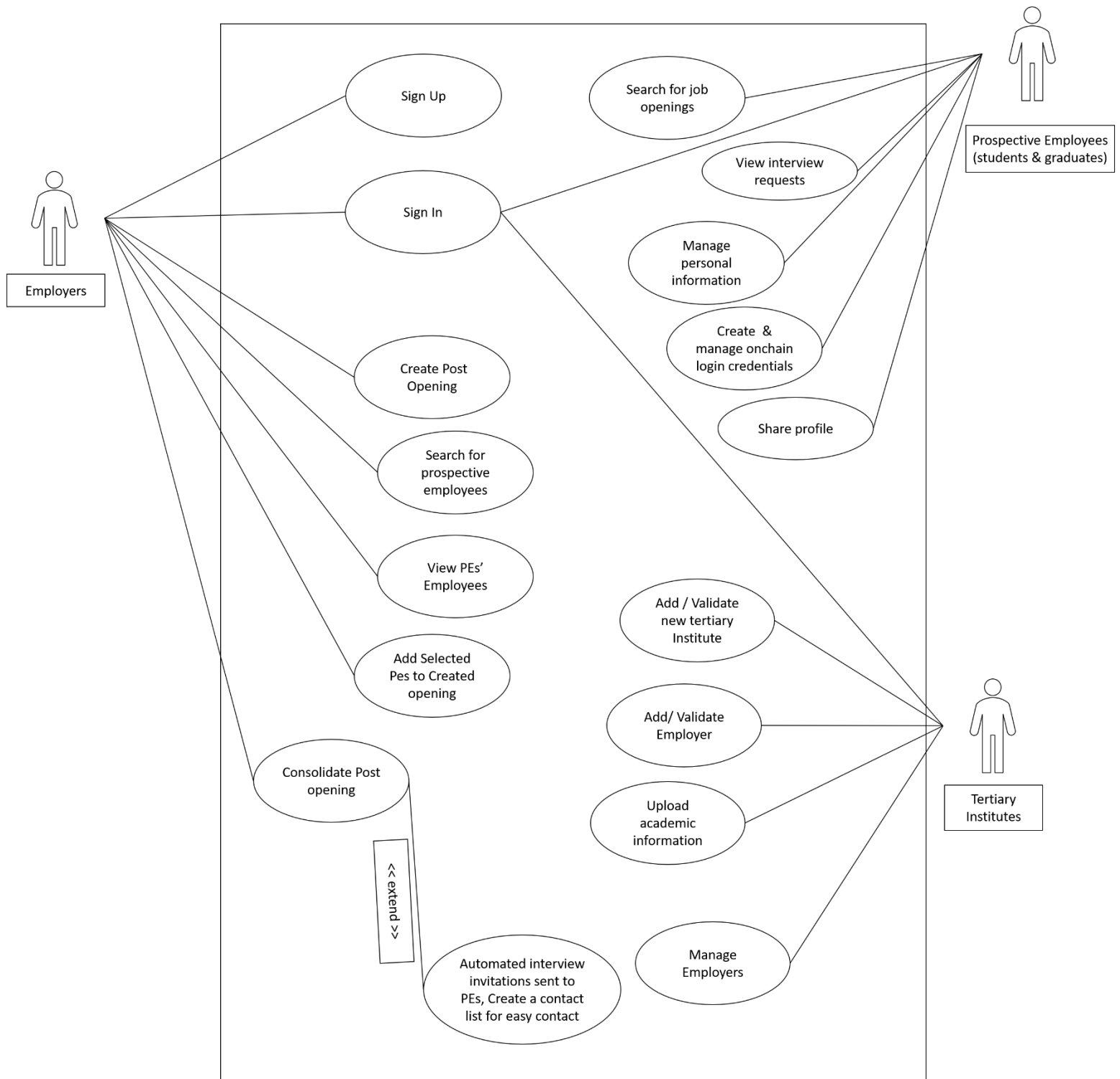


Figure 8: 3.6 User Case Diagram for Proposed System

## 4 Chapter Four –Design

### 4.1 Systems Diagrams

#### 4.1.1 Context diagrams

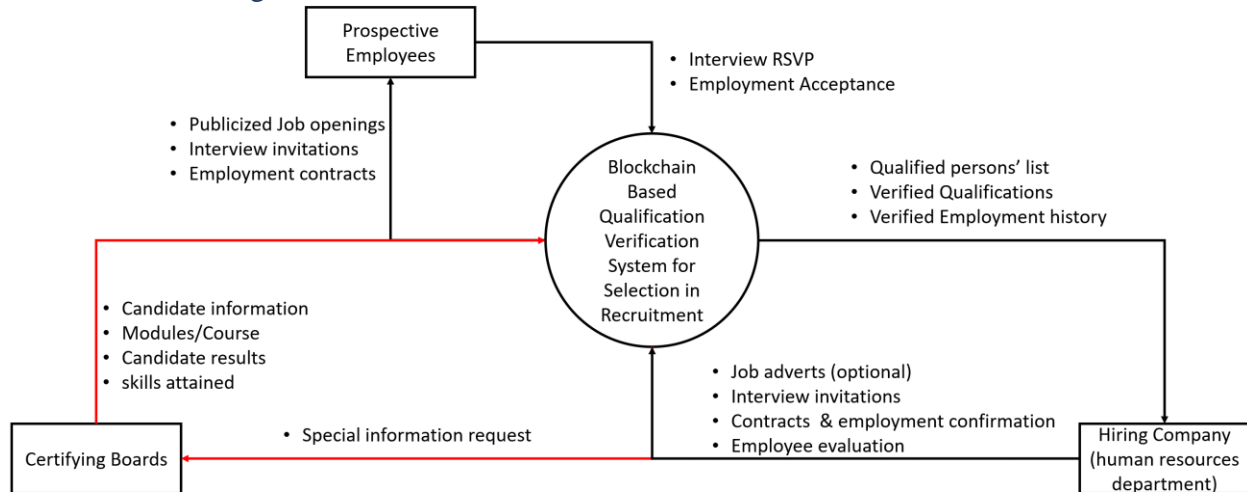
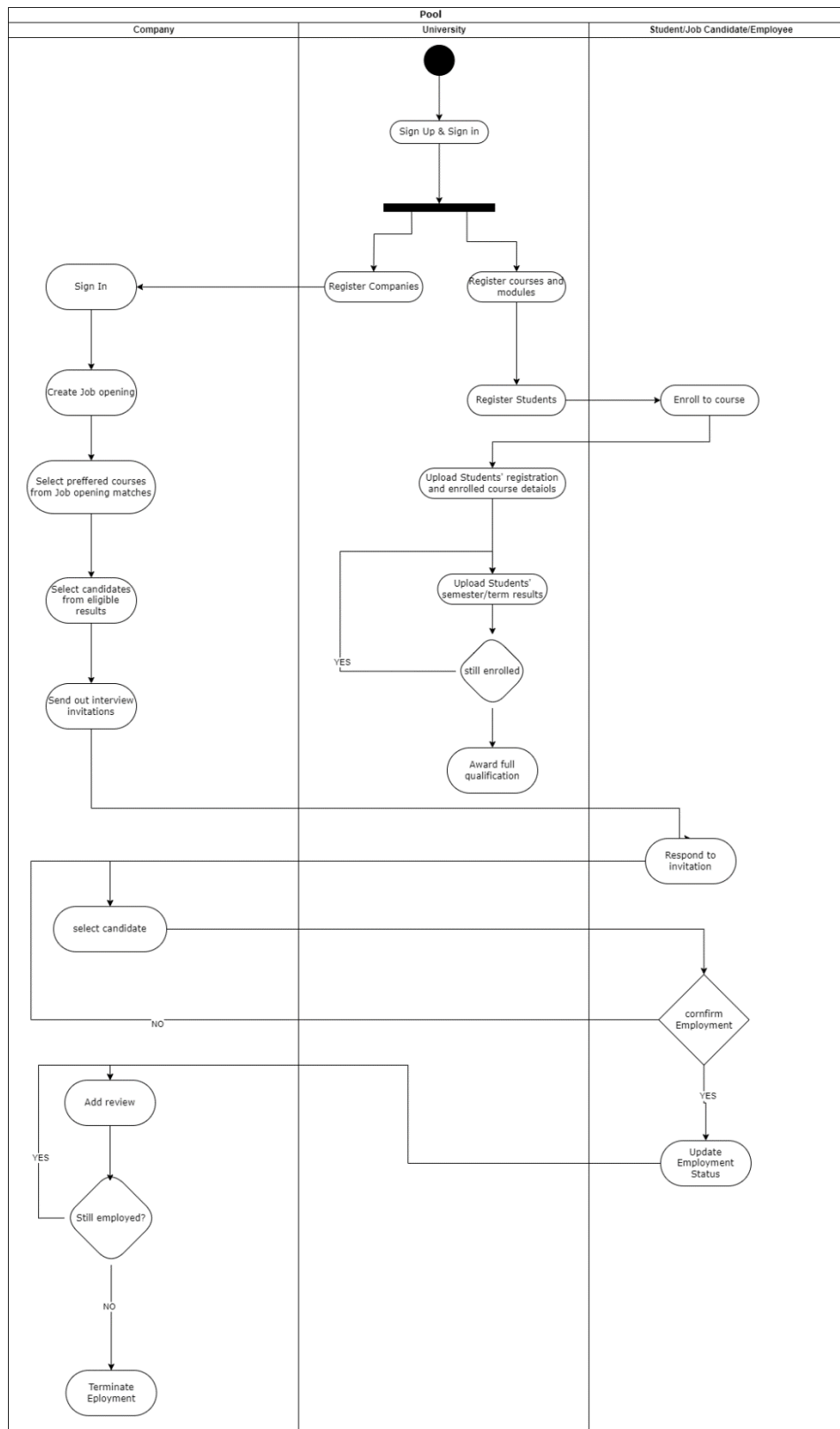


Figure 9: Context Diagram of Proposed Sytem

## 21



### 4.1.3 Activity diagrams



*Figure 11: Activity Diagram*

## 4.2 Architectural Design

The client-side of the web app is implemented using HTML, CSS, and JavaScript. The user interacts with the client-side of the web app using a web browser. The client-side of the web app interacts with the server-side of the web app using HTTP requests.

The server-side of the web app is responsible for interacting with the Ethereum network and smart contracts. It is implemented using a backend Python flask as the backend. The server-side of the web app interacts with the smart contracts deployed on the Ethereum network using a web3.py library.

The web app uses the following architecture:

1. **User Interface Layer:** The user interface layer is the client-side of the web app. It consists of HTML, CSS, and JavaScript. The user interacts with the web app using a web browser. The user interface layer sends HTTP requests to the server-side of the web app.
2. **Application Layer:** The application layer is the server-side of the web app. It consists of a web server and an Ethereum client. The web server receives HTTP requests from the user interface layer and processes them. The Ethereum client interacts with the Ethereum network and smart contracts.
3. **Smart Contract Layer:** The smart contract layer consists of the smart contracts deployed on the Ethereum network. The server-side of the web app interacts with the smart contract layer using the web3.py library.
4. **Data Layer:** The data layer consists of the data stored on the Ethereum network. The data is stored in a decentralized manner on the Ethereum network.

The architecture allows the web app to interact with the Ethereum network and smart contracts in a secure and decentralized manner.

### 4.2.1 Hardware

The blockchain recruitment system will be hosted on a cloud-based server infrastructure with a load balancer to ensure high availability and scalability. The servers will be equipped with high-performance processors and memory, and the data will be stored on a distributed blockchain network to ensure redundancy and data integrity.

#### 4.2.2 Networking

The architecture of a blockchain system with a web-based interface that connects to the Ethereum network via web3.py is a client-server model with the web server acting as the client and the Ethereum network and smart contracts acting as the server. The web server communicates with the Ethereum network through web3.py, and smart contracts enforce the rules and logic of the system. The database stores the data related to the blockchain system, and an API allows other applications to interact with the blockchain.

### 4.3 Database Design

#### 4.3.1 Normalized Databases

*Table 3: Companies Table*

Field	Data Type	Key
address	address	Primary
company_name	string	
physical_address	string	
cell	string	
email	string	
telephone	string	
added_by	address	Foreign

*Table 4: Job opening candidate choices table.*

Field	Data Type	Key
candidate	address	
university	address	
course	string	
company_address	address	Foreign
job_opening_index	uint	Foreign

Table 5: job openings table

Field	Data Type	Key
company_address	address	Primary
job_opening_index	uint	
job_opening_title	string	
job_opening_description	string	
job_opening_location	string	
job_opening_salary	string	
job_opening_status	string	
job_opening_start_date	string	
job_opening_interview_date	string	
job_opening_choices_count	uint	
job_opening_field	string	

Table 6: Personal skills

Field	Data Type	Key
address	address	Foreign
name	string	
proficiency	uint	

Table 7: universities

Field	Data Type	Key
address	address	Primary
uni_name	string	
physical_address	string	
cell	string	
email	string	

telephone	string	
added_by	address	Foreign

*Table 8: university courses*

<b>Field</b>	<b>Data Type</b>	<b>Key</b>
id	int	Primary
uni_address	address	Foreign (universities)
course_code	string	
course_name	string	
course_description	string	
course_duration	string	
course_field	string	
course_level	string	
course_status	string	
course_start_date	string	
course_end_date	string	
added_by	address	Foreign (universities)



### 4.3.2 ER diagrams.

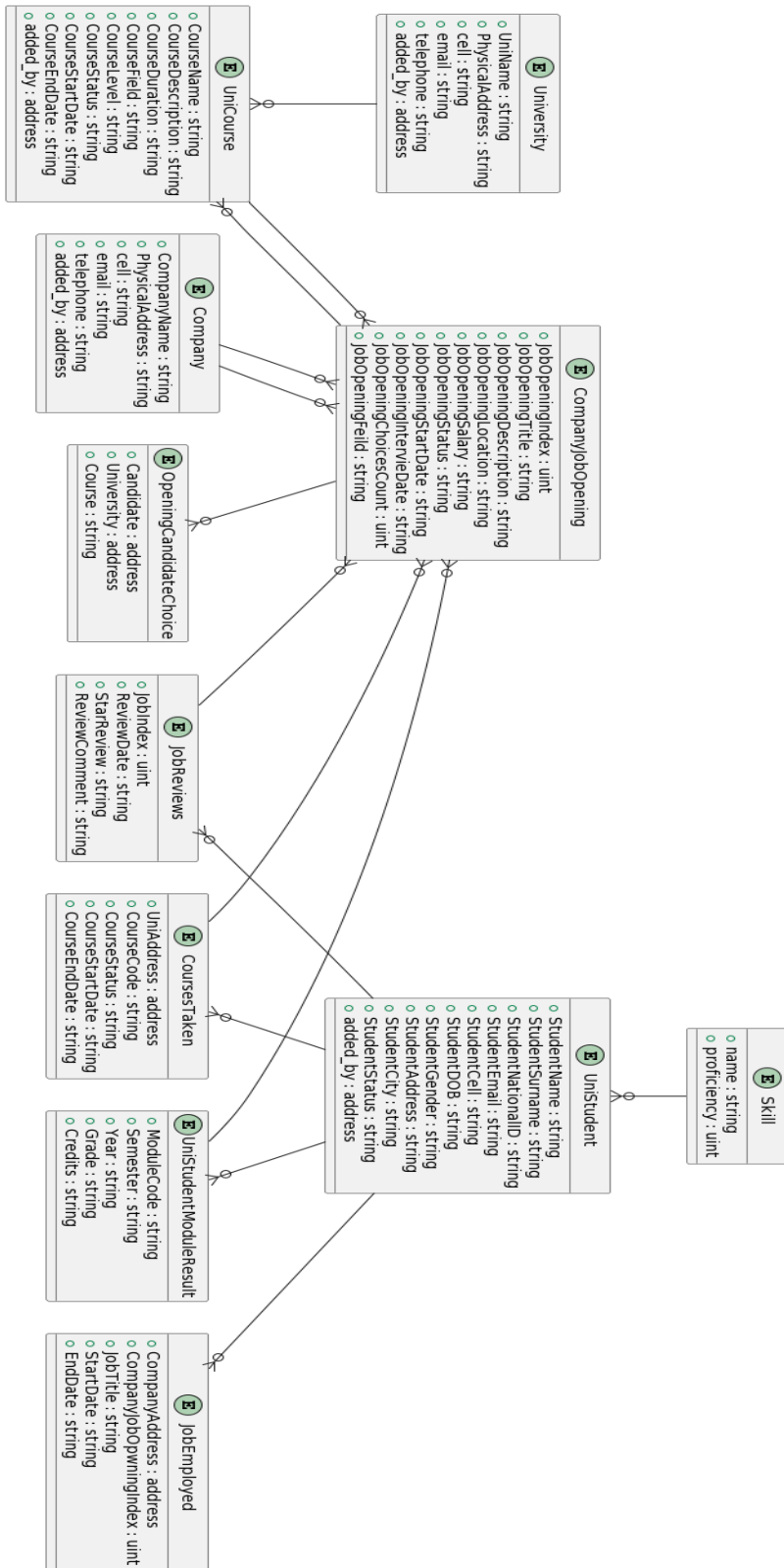


Figure 12: ERD

## 4.4 Program Design

### 4.4.1 Class diagrams

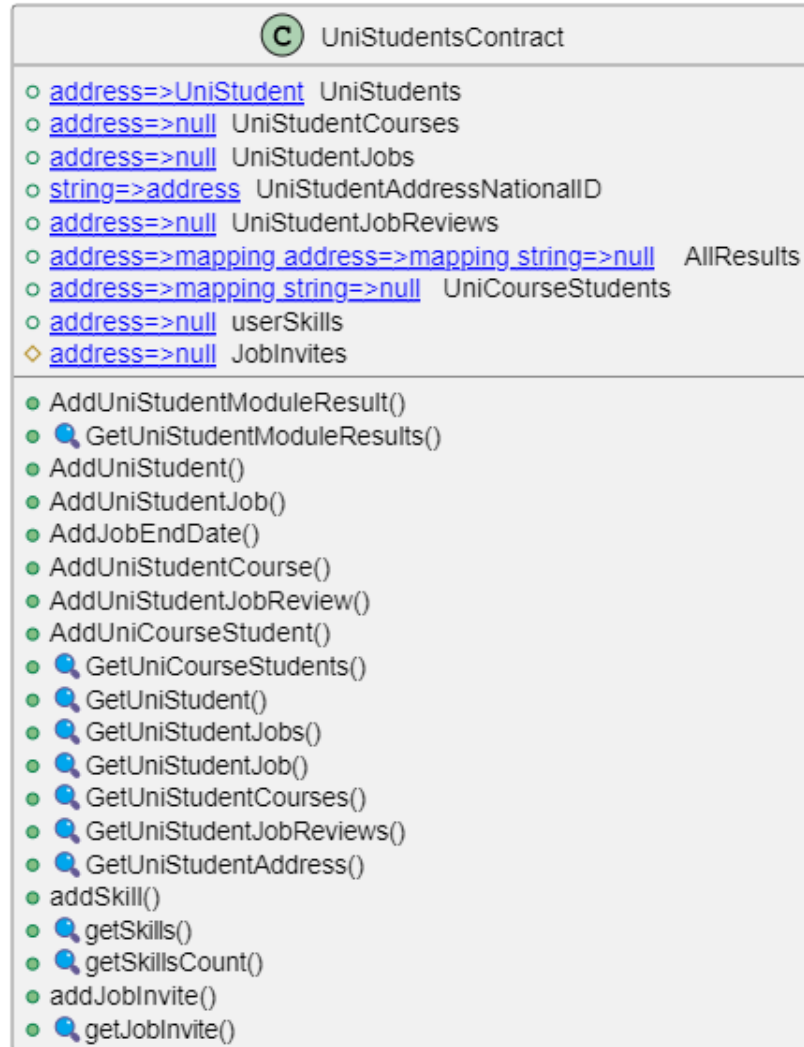




























Figure 13: Class Diagrams

C UniInformation	
○ <a href="#">address=&gt;University</a>	Universities
○ <a href="#">address</a>	UniAccts
<ul style="list-style-type: none"> <li>● addUni()</li> <li>●  getUni()</li> <li>●  getAllUnis()</li> </ul>	

C CompanyContract	
○ <a href="#">address=&gt;Company</a>	Companies
○ <a href="#">address=&gt;mapping_uint=&gt;null</a>	JobPreferredQualifications
○ <a href="#">address=&gt;null</a>	CompanyJobOpenings
○ <a href="#">address=&gt;mapping_uint=&gt;null</a>	JobPreferredSkills
○ <a href="#">address=&gt;mapping_uint=&gt;null</a>	JobOpeningCandidates
○ <a href="#">address=&gt;mapping_uint=&gt;null</a>	JobOpeningChoices
○ <a href="#">address=&gt;mapping_uint=&gt;null</a>	JobOpeningRSVP
○ <a href="#">address=&gt;null</a>	CompanyEmployees
<ul style="list-style-type: none"> <li>● AddCompany()</li> <li>●  GetCompanyByAddress()</li> <li>● addJobOpeningRSVP()</li> <li>●  getJobOpeningRSVP()</li> <li>● AddCompanyJobOpening()</li> <li>● ChangeJobOpeningStatus()</li> <li>● AddJobPreferredQualifications()</li> <li>● AddJobPreferredSkills()</li> <li>● AddJobOpeningCandidates()</li> <li>● AddJobOpeningChoices()</li> <li>●  GetJobOpeningCandidates()</li> <li>●  GetJobOpeningChoices()</li> <li>●  GetJobPreferredQualifications()</li> <li>●  GetJobPreferredSkills()</li> <li>●  GetCompanyJobOpenings()</li> <li>●  GetCompanyJobOpeningsCount()</li> <li>●  GetCompanyJobOpening()</li> <li>●  GetCompanyEmployees()</li> <li>● RemoveEmployee()</li> <li>●  GetCompanyEmployeesCount()</li> <li>●  GetCompanyEmployee()</li> </ul>	

C UniCoursesContract	
○ <a href="#">address=&gt;null</a>	UniCoursesList
○ <a href="#">address=&gt;mapping_string=&gt;UniCourse</a>	UniCourses
○ <a href="#">address=&gt;mapping_string=&gt;null</a>	UniCourseModules
○ <a href="#">address=&gt;mapping_string=&gt;null</a>	UniCoursePrerequisites
○ <a href="#">address=&gt;mapping_string=&gt;null</a>	UniCourseJobs
○ <a href="#">address=&gt;mapping_string=&gt;null</a>	UniCourseTasks
○ <a href="#">address=&gt;null</a>	UniModulesList
○ <a href="#">address=&gt;mapping_string=&gt;Module</a>	UniModules
○ <a href="#">address=&gt;mapping_string=&gt;null</a>	UniModuleSkills
○ <a href="#">address=&gt;mapping_string=&gt;null</a>	UniModuleJobs
○ <a href="#">address=&gt;mapping_string=&gt;null</a>	UniModuleTasks
○ <a href="#">address=&gt;mapping_string=&gt;null</a>	UniModuleSkillCategories
<ul style="list-style-type: none"> <li>● AddUniCourse()</li> <li>● AddUniCourseModule()</li> <li>● AddUniCoursePrerequisite()</li> <li>● AddUniCourseJob()</li> <li>● AddUniCourseTask()</li> <li>●  GetUniCourseTasks()</li> <li>●  GetUniCourseModules()</li> <li>●  GetUniCoursePrerequisites()</li> <li>●  GetUniCourseJobs()</li> <li>●  GetUniCourse()</li> <li>●  GetAllUniCourses()</li> <li>● AddUniModuleSkillCategory()</li> <li>●  GetUniModuleSkillCategories()</li> <li>● AddUniModule()</li> <li>● ChangeCategoryStatus()</li> <li>● AddUniModuleSkill()</li> <li>● AddUniModuleJob()</li> <li>● AddUniModuleTask()</li> <li>●  GetUniModuleTasks()</li> <li>●  GetUniModuleSkills()</li> <li>●  GetUniModuleJobs()</li> <li>●  GetUniModule()</li> <li>●  GetUniModules()</li> </ul>	

#### 4.4.2 Sequence diagrams

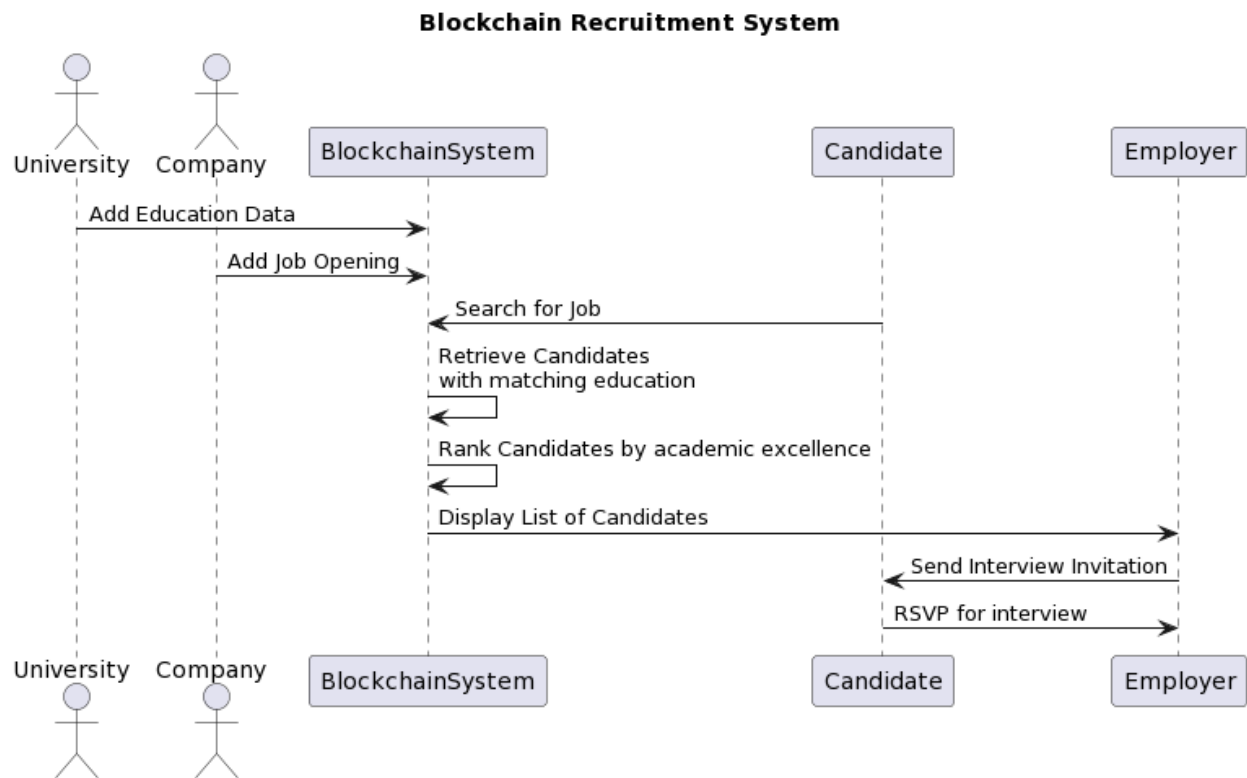


Figure 14: Sequence Diagram

#### 4.4.3 Package diagrams

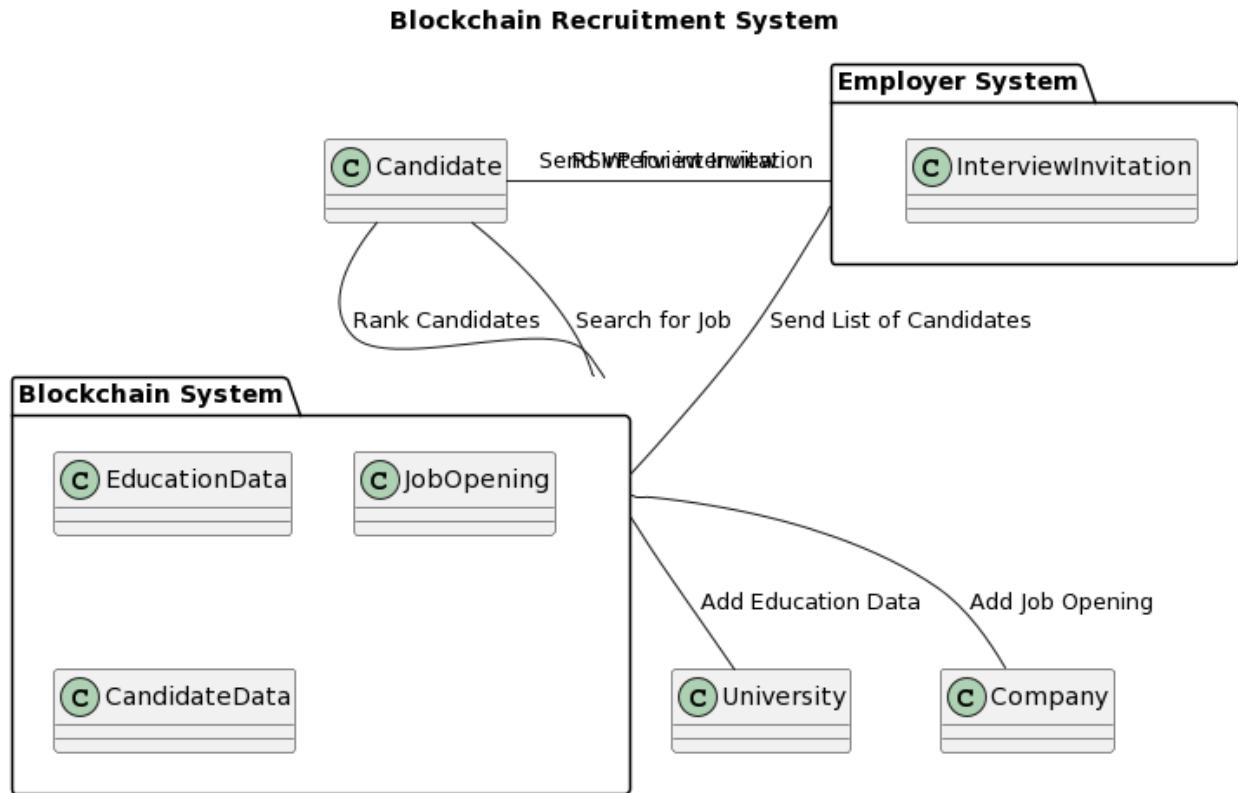


Figure 15: Package Diagram

#### 4.4.4 Pseudo code

1. University enters education data into system
2. Company enters job opening into system
3. Company searches for job candidates based on education
4. System retrieves all candidates with matching education
5. System ranks candidates by academic excellence
6. System displays list of candidates to employer
7. Employer sends interview invitation to candidate
8. Candidate RSVPs for interview

## 4.5 Interface Design-Screenshots of user interface

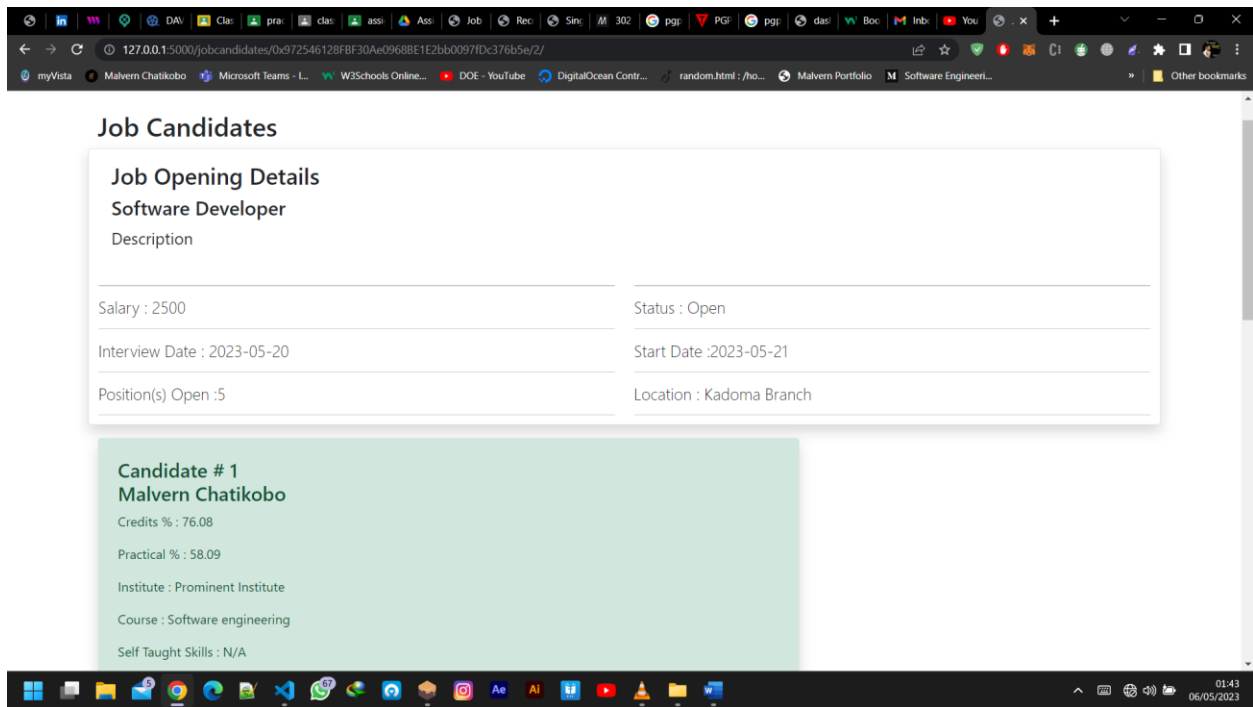


Figure 16: Job Candidates Page

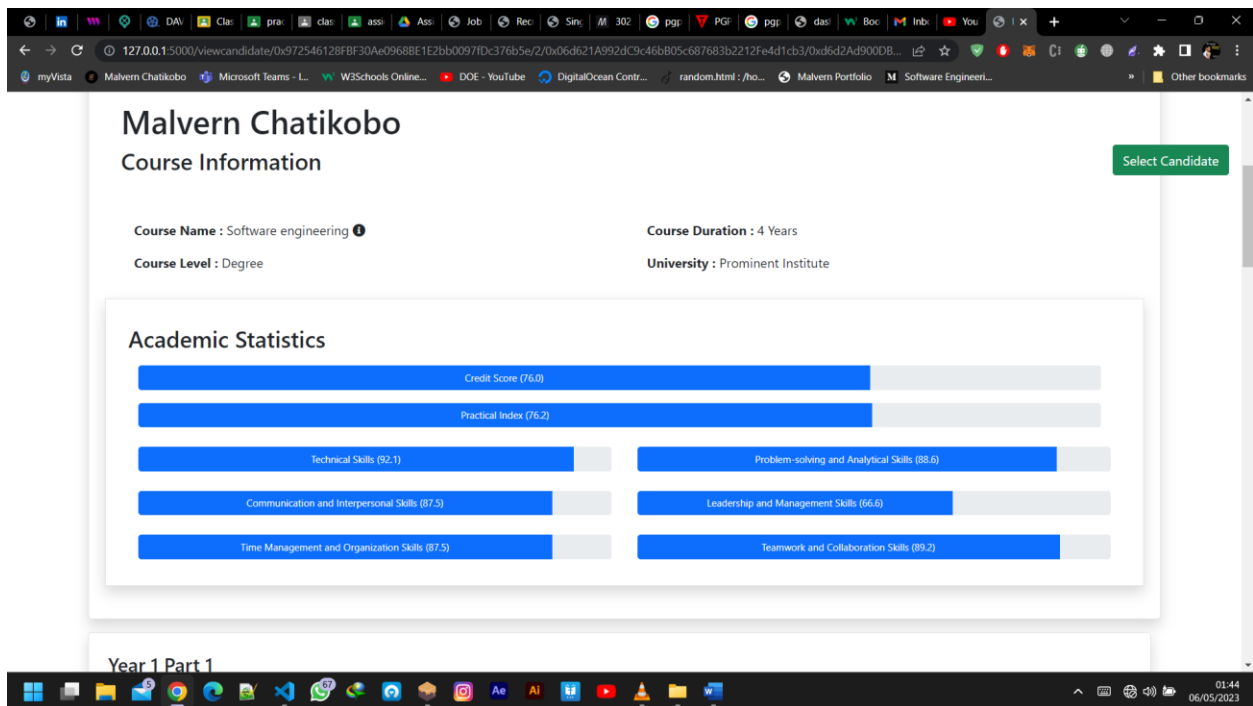


Figure 17: Candidate Profile 1

Module Name	Module Grade	Practical
Technopreneurship III	2.2	
Computer Architecture & Organisation	2.2	✓
Advanced Databases	2.2	✓
Data Communications and Networks	2.1	✓
Software Project Management		
Operations Research	1	
Object Oriented Programming in Java	1	✓

**Year 3 Part 2**

Module Name	Module Grade	Practical
Internship	2.1	✓

**Year 3 Part 1**

Module Name	Module Grade	Practical
Operations Research	1	

*Tooltip for Data Communications and Networks: This module provides students with an understanding of data communications and networks. It covers topics such as data transmission, data encoding, data link control, and network protocols. Students will learn about the different types of networks and how they work. They will also learn about the different components of a network and how they interact with each other.*

Figure 18: Candidate Profile 2

## Add Course

Course Name

Course Code

Course Description

Course Duration

Course Duration Weeks

Select Course Field :  
Aeronautics and Aviation Science

Select Course Level:  
Certificate

Course Start Date  
dd/mm/yyyy

Course Prerequisites (one per line)

Jobs the Course is directly applicable in (one per line)

Save

Figure 19: Add Course Page

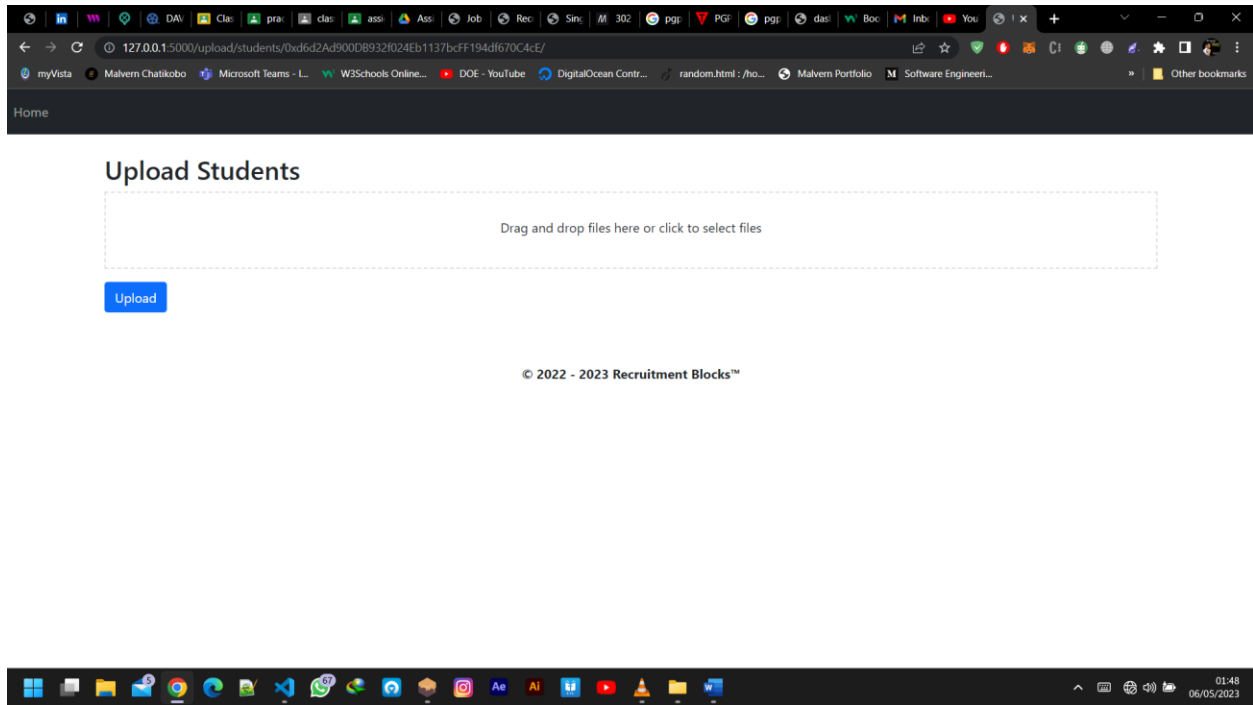


Figure 20: Upload Students Page

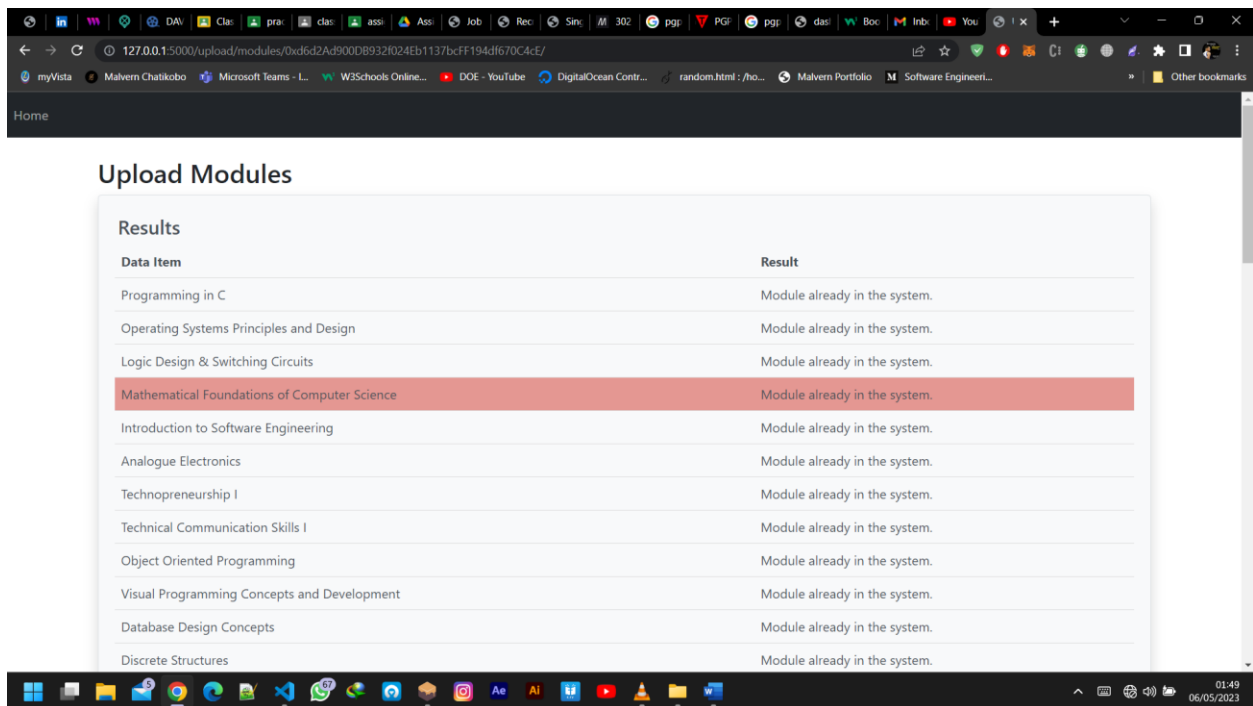


Figure 21: Upload Module Results



## 5 Chapter Five-Implementation & Testing

### 5.1 Software Testing-

#### 5.1.1 Unit Testing

##### 5.1.1.1 Testing the University Information Smart Contract

```
var UniInformation = artifacts.require("c:\Users\Lenovo Yoga\Desktop\HIT 400  
Detailed Pleasure\contracts\UniInformation.sol");

contract('UniInformation', (accounts) => {
  var creatorAddress = accounts[0];
  var firstOwnerAddress = accounts[1];
  var secondOwnerAddress = accounts[2];
  var externalAddress = accounts[3];
  var unprivilegedAddress = accounts[4]
  /* create named accounts for contract roles */

  it('should revert if ...', () => {
    return UniInformation.deployed()
      .then(instance => {
        return instance.publicOrExternalContractMethod(argument1,  
argument2, {from:externalAddress});
      })
      .then(result => {
        assert.fail();
      })
      .catch(error => {
        assert.notEqual(error.message, "assert.fail()", "Reason ...");
      });
  });

  context('testgroup - security tests - description...', () => {
    //deploy a new contract
    before(async () => {
      /* before tests */
      const newUniInformation = await UniInformation.new()
    })

    it('fails on initialize ...', async () => {
      return assertRevert(async () => {
        await newUniInformation.initialize()
      })
    })
  })
})
```

```

    })
  })

  it('checks if method returns true', async () => {
    assert.isTrue(await newUniInformation.thisMethodShouldReturnTrue())
  })
})
});

```

#### 5.1.1.2 Testing Uni Student Contract

```

pragma solidity >=0.4.25 <0.9.0;
import "remix_tests.sol"; // this import is automatically injected by Remix.
import "./UniStudentsContract.sol"; // this import assumes that the
UniStudentsContract.sol file is in the same directory as this test file.

contract UniStudentsContractTest {
  UniStudentsContract uniStudentsContract;

  function beforeEach() public {
    uniStudentsContract = new UniStudentsContract();
  }

  function testAddUniStudent() public {
    // Arrange
    address studentAddress = address(this);
    string memory studentName = "John";
    string memory studentSurname = "Doe";
    string memory studentNationalID = "1234567890";
    string memory studentEmail = "john.doe@example.com";
    string memory studentCell = "1234567890";
    string memory studentDOB = "1990-01-01";
    string memory studentGender = "Male";
    string memory studentAddress1 = "123 Main St";
    string memory studentCity = "Los Angeles";
    string memory studentStatus = "Part-time";

    // Act
    string memory result = uniStudentsContract.AddUniStudent(studentAddress,
studentName, studentSurname, studentNationalID, studentEmail, studentCell,
studentDOB, studentGender, studentAddress1, studentCity, studentStatus);

    // Assert

```

```

        Assert.equal(result, "Student Added", "AddUniStudent should return
'Student Added'");
        Assert.equal(uniStudentsContract.UniStudents(studentAddress).StudentName,
studentName, "Student name should match");
        Assert.equal(uniStudentsContract.UniStudents(studentAddress).StudentSurname,
studentSurname, "Student surname should match");
        Assert.equal(uniStudentsContract.UniStudents(studentAddress).StudentNationalID,
studentNationalID, "Student national ID should match");
        Assert.equal(uniStudentsContract.UniStudents(studentAddress).StudentEmail,
studentEmail, "Student email should match");
        Assert.equal(uniStudentsContract.UniStudents(studentAddress).StudentCell,
studentCell, "Student cell should match");
        Assert.equal(uniStudentsContract.UniStudents(studentAddress).StudentDOB,
studentDOB, "Student date of birth should match");
        Assert.equal(uniStudentsContract.UniStudents(studentAddress).StudentGender,
studentGender, "Student gender should match");
        Assert.equal(uniStudentsContract.UniStudents(studentAddress).StudentAddress,
studentAddress1, "Student address should match");
        Assert.equal(uniStudentsContract.UniStudents(studentAddress).StudentCity,
studentCity, "Student city should match");
        Assert.equal(uniStudentsContract.UniStudents(studentAddress).StudentStatus,
studentStatus, "Student status should match");
    }
}

```

### 5.1.2 Module Testing

#### 1. User registration and profile creation:

- Test: Create a new user account and ensure that the user's details are correctly stored in the database.
- Test: Attempt to create a user account with invalid or incomplete information and ensure that appropriate error messages are displayed.

#### 2. Educational institution profile creation:

- Test: Create a new educational institution profile and ensure that the details are correctly stored in the database.
- Test: Attempt to create an educational institution profile with invalid or incomplete information and ensure that appropriate error messages are displayed.

#### 3. Job posting creation:

- Test: Create a new job posting and ensure that the details are correctly stored in the database.
  - Test: Attempt to create a job posting with invalid or incomplete information and ensure that appropriate error messages are displayed.
4. Candidate matching:
- Test: Attempt to match a job posting with a candidate's educational background and ensure that the matching algorithm is working correctly.
  - Test: Create a job posting with a specific educational requirement and ensure that the system only displays candidates who meet that requirement.
5. Ranking of candidates based on academic excellence:
- Test: Create a job posting with a specific educational requirement and ensure that the system ranks candidates in order of academic excellence.
  - Test: Ensure that the system takes into account other factors (such as work experience) that may affect a candidate's suitability for the job.
6. Interview scheduling:
- Test: Create a new interview invitation and ensure that the details are correctly stored in the database.
  - Test: Attempt to create an interview invitation with invalid or incomplete information and ensure that appropriate error messages are displayed.
7. RSVP functionality:
- Test: Create an interview invitation and ensure that the candidate is able to RSVP using the link provided.
  - Test: Ensure that the system sends reminder emails to candidates who have not yet RSVP'd.
8. Blockchain integration:
- Test: Ensure that all data is correctly stored on the blockchain system and that it can be retrieved and verified as needed.
  - Test: Attempt to tamper with data stored on the blockchain and ensure that the system detects and flags any unauthorized changes.

### 5.1.3 Integration Testing

1. Posting education information
  - Test that a university can post education information for an individual.
  - Test that the education information is correctly stored on the blockchain.
  - Test that the education information can be retrieved from the blockchain.
2. Adding job opening information
  - Test that a company can add a job opening to the system.
  - Test that the job opening information is correctly stored on the blockchain.
  - Test that the job opening information can be retrieved from the blockchain.
3. Matching candidates to jobs
  - Test that the system can match candidates to job openings based on education.
  - Test that the matching algorithm puts candidates in order of academic excellence.
4. Sending interview invitations
  - Test that the system can send interview invitations to candidates.
  - Test that the links provided in the invitations work correctly.
5. Storing interview RSVPs
  - Test that the system can store information about interview RSVPs on the blockchain.
  - Test that the RSVP information can be retrieved from the blockchain.

### 5.1.4 System Testing

- Test the overall functionality of the system, including its ability to match candidates with appropriate job openings, store information on a blockchain system, and facilitate communication between employers and candidates.
- Test the system's user interface and user experience, ensuring that it is user-friendly and intuitive.
- Test the system's performance under heavy loads, ensuring that it can handle a large number of users and transactions without crashing or slowing down.

### 5.1.5 Database Testing

- Test the integrity and accuracy of the database, ensuring that information is stored correctly and can be retrieved accurately.
- Test the system's ability to handle large amounts of data, ensuring that it can scale to meet the needs of a growing user base.

### 5.1.6 Acceptance Testing

- I tested the system against the requirements and specifications provided.
- I tested whether the system meets the needs of both universities and companies, providing accurate and relevant information to both parties.
- Ensure that the system is easy to use and navigate, and that it provides a positive user experience for all users.

Sample code

## 5.2 Sample Code

### 5.2.1 Universities Smart Contract

```
pragma solidity >=0.4.25 <0.9.0;

contract UniInformation {

    struct University { // Struct
        string UniName;
        string PhysicalAddress;
        string cell;
        string email;
        string telephone;
        address added_by;

    }

    mapping (address=>University) public Universities;

    address[] public UniAccts;

    function addUni(address _UniAccount,string memory _UniName, string memory
    _PhysicalAddress, string memory _cell, string memory _email, string memory
    _telephone) public returns(string memory) {
        if(UniAccts.length == 0) {
```

```

        Universities[_UniAccount] = University(_UniName, _PhysicalAddress,
        _cell, _email, _telephone, msg.sender);
        UniAccts.push(_UniAccount) ;
        return "University added";
    } else {
        for(uint i = 0; i < UniAccts.length; i++) {
            if(UniAccts[i] == msg.sender) {
                Universities[_UniAccount] = University(_UniName,
        _PhysicalAddress, _cell, _email, _telephone, msg.sender);
                UniAccts.push(_UniAccount) ;
                return "University added";
            }
            else {
                return "University not added";
            }
        }
    }
}

function getUni(address _UniAccount) public view returns (string memory,
string memory, string memory, string memory, address) {
    return (Universities[_UniAccount].UniName,
Universities[_UniAccount].PhysicalAddress, Universities[_UniAccount].cell,
Universities[_UniAccount].email, Universities[_UniAccount].telephone,
Universities[_UniAccount].added_by);
}

function getAllUnis() public view returns(address[] memory){
    return (UniAccts);
}
}

```

### 5.2.2 Search and Ranking Functions

```

def getCourseTotalCredits(uni,course):
    modules=TheEdu.GetAUniCourseModules(uni,course)
    total_credits=0
    for module in modules:
        total_credits+=TheEdu.GetAUniModule(uni,module)[2]
    return total_credits

```

```

#for the initial search using the job title using course jobs
def StudentsPerCourse(job,company, jobid):
    candidates=TheComp.GetJobOpeningCandidates(company, jobid)
    riz={}
    pple=[]
    for a in candidates:
        pple.append(a[0])

    rez_jobs=searchCourseJobs(job)
    print("1"+ str(rez_jobs))
    if rez_jobs!={}:
        for uni in rez_jobs:
            for course in rez_jobs[uni]:

                #riz=PeopleSearch(uni,course,company, jobid)

                rez_1={}
                students_percentages={} #the persons credit % for a course
                print("#####")
                print(Munhu.GetAUniCourseStudents(course, uni))
                print("#####")

                for person in Munhu.GetAUniCourseStudents(course, uni):
                    print("munhu "+person)
                    if person in pple:
                        continue
                    total_credits_recieved=0
                    total_credits_possibble=0
                    total_practical_credits=0
                    persons_module_results=Munhu.GetAUniStudentModuleResults(per
on, uni, course)

                    print("#####TRRRRRRRRRRRRRRRRRRRRRRR#####")

                    print(persons_module_results)

                    print("#####")
                    for module in persons_module_results:
                        total_credits_recieved+=float(module[4])

```



```

        the_mod=TheEdu.GetAUniModule(uni, module[0])
        total_credits_possibble+=float(the_mod[2])
        if the_mod[6]==True:
            total_practical_credits+=float(module[4])
        nm=Munhu.GetAUniStudent(person)
        print("Total recieved : " + str(total_credits_recieved))
        print("Total possible : " + str(total_credits_possibble))
        print("Total practical : " + str(total_practical_credits))

        rez_1[person]={ 'TotalCreditsRecieved':total_credits_recieved,
'TotalCreditsPossibble':total_credits_possibble, 'TotalPracticalCredits':
total_practical_credits, 'TotalPracticalPercentage':
(total_practical_credits/total_credits_recieved)*100,
'Percentage':(total_credits_recieved/total_credits_possibble)*100,
'University':uni, 'Course':course, 'UniversityName':TheUni.getUniversity(uni)[0],
'CourseName':TheEdu.GetAUniCourse(uni,
course)[0], 'SelfTaughtCount':Munhu.getSkillsCount(person), 'Name':nm[0]+' '+nm[1]
}

        if int((total_credits_recieved/total_credits_possibble)*100)
in students_percentages:
            students_percentages[int((total_credits_recieved/total_cr
edits_possibble)*100)].append(person)
        else:
            students_percentages[int((total_credits_recieved/total_cr
edits_possibble)*100)]=[person]

        rez_1['SortedStudents']=dict(sorted(students_percentages.items(),
reverse=True))

    return rez_1
else:
    rez_jobs=searchModuleJobs(job)
    print(rez_jobs)
    for x in rez_jobs['SortedCourseCounts']:
        for y in rez_jobs['SortedCourseCounts'][x]:
            uni=y['University']
            course=y['Course']
            #riz=PeopleSearch(y['University'],y['Course'],company, jobid)

    rez_1={}
    students_percentages={} #the persons credit % for a course
    for person in Munhu.GetAUniCourseStudents(course, uni):

```

```

        #print("munhu "+person)
        if person in pple:
            continue
        total_credits_recieved=0
        total_credits_possibble=0
        total_practical_credits=0
        persons_module_results=Munhu.GetAUniStudentModuleResults(pers
on, uni, course)

        #print("person module results"+str(persons_module_results))
        for module in persons_module_results:
            total_credits_recieved+=float(module[4])
            the_mod=TheEdu.GetAUniModule(uni, module[0])
            total_credits_possibble+=float(the_mod[2])
            if the_mod[6]==True:
                total_practical_credits+=float(module[4])
            rez_1[person]={ 'TotalCreditsRecieved':total_credits_recieved,
'TotalCreditsPossibble':total_credits_possibble, 'TotalPracticalCredits':
total_practical_credits, 'TotalPracticalPercentage':
(total_practical_credits/total_credits_recieved)*100,
'Percentage':(total_credits_recieved/total_credits_possibble)*100,
'University':uni, 'Course':course, 'UniversityName':TheUni.getUniversity(uni)[0],
'CourseName':TheEdu.GetAUniCourse(uni, course)[0]}
            if int((total_credits_recieved/total_credits_possibble)*100)
in students_percentages:
                students_percentages[int((total_credits_recieved/total_cr
edits_possibble)*100)].append(person)
            else:
                students_percentages[int((total_credits_recieved/total_cr
edits_possibble)*100)]=[person]
            print("%"+str(students_percentages))

        rez_1['SortedStudents']=dict(sorted(students_percentages.items(),
reverse=True))

        print('Rez1'+ str(rez_1))

        return rez_1

    return riz

def PeopleSearch(uni, course,company, jobid):
    candidates=TheComp.GetaJobOpeningCandidates(company, jobid)

```

```

pple=[]
for a in candidates:
    pple.append(a[0])

rez_1={}
students_percentages={} #the persons credit % for a course
for person in Munhu.GetAUniCourseStudents(course, uni):
    print("munhu "+person)
    print("Vanhu "+pple)

    if person in pple:
        continue
    total_credits_recieved=0
    total_credits_possibble=0
    total_practical_credits=0
    persons_module_results=Munhu.GetAUniStudentModuleResults(person, uni,
course)
    #print("person module results"+str(persons_module_results))
    for module in persons_module_results:
        total_credits_recieved+=float(module[4])
        the_mod=TheEdu.GetAUniModule(uni, module[0])
        total_credits_possibble+=float(the_mod[2])
        if the_mod[6]==True:
            total_practical_credits+=float(module[4])
        rez_1[person]={ 'TotalCreditsRecieved':total_credits_recieved,
'TotalCreditsPossibble':total_credits_possibble, 'TotalPracticalCredits':
total_practical_credits, 'TotalPracticalPercentage':
(total_practical_credits/total_credits_recieved)*100,
'Percentage':(total_credits_recieved/total_credits_possibble)*100,
'University':uni, 'Course':course, 'UniversityName':TheUni.getUniversity(uni)[0],
'CourseName':TheEdu.GetAUniCourse(uni, course)[0]}
        if int((total_credits_recieved/total_credits_possibble)*100) in
students_percentages:
            students_percentages[int((total_credits_recieved/total_credits_possib
ble)*100)].append(person)
        else:
            students_percentages[int((total_credits_recieved/total_credits_possib
ble)*100)]=[person]
        print("%"+str(students_percentages))

    rez_1['SortedStudents']=dict(sorted(students_percentages.items(),
reverse=True))

```

```
return rez_1
```

## 6 Chapter Six –Conclusions and Recommendations

### 6.1 Results and summary

Overall, the system performed well in allowing universities and tertiary institutions to post education information, companies to add job opening information, and finding candidates with the appropriate education for the job. The system was also able to sort the candidates based on academic excellence and provide employers with the ability to send out interview invitations with links for candidates to RSVP. The blockchain storage system also provided secure storage for all information.

### 6.2 Recommendations

Based on the testing results, some recommendations for the system include:

1. Improving the user interface to make it more intuitive and user-friendly for both employers and candidates.
2. Adding more advanced search filters for employers to narrow down the pool of candidates even further.
3. Integrating with popular job search platforms to increase visibility and attract more candidates.

### 6.3 Future Works

Some potential areas of future work for the system could include:

1. Expanding the system to include additional types of educational institutions, such as vocational schools or certification programs.
2. Integrating with HR software to provide a seamless hiring process for employers.
3. Adding machine learning algorithms to improve the matching process between candidates and job openings.
4. Incorporating more advanced security measures for the blockchain storage system, such as multi-factor authentication and encryption.

## 7 Bibliography

### Appendices

#### 7.1 References

- [1] Mohammed Elhazzam (2015) The Effect of ICT on Human Resources Management Practices Case of Number of Organizations in Southwest Algeria (Bechar City). International Journal of Innovative Research in Engineering & Management (IJIREM) ISSN: 2350-0557, Volume-2, Issue-3, May-2015
- [2] Halpern, C. and Halpern, B. (2018) Diversions: Obstacles for Undergraduate Students when Applying for Internship Positions Online. Open Journal of Business and Management, 6, 165-182. doi: 10.4236/ojbm.2018.61011.
- [3] Tzu-Yu Yeh, Rasha Kashef (2020) Trust-Based Collaborative Filtering Recommendation Systems on the Blockchain. Advances in Internet of Things, Vol.10 No.4, 2020
- [4] A. A. Monrat, O. Schelén and K. Andersson, "A Survey of Blockchain from the Perspectives of Applications, Challenges, and Opportunities," in IEEE Access, vol. 7, pp. 117134-117151, 2019, doi: 10.1109/ACCESS.2019.2936094.
- [5] Rozario, S.D.; Venkatraman, S.; Abbas, A. Challenges in Recruitment and Selection Process: An Empirical Study. Challenges 2019, 10, 35. <https://doi.org/10.3390/challe10020035>
- [6] Chungyalpa W, Karishma T (2016) Best Practices and Emerging Trends in Recruitment and Selection. J Entrepren Organiz Manag 5: 173. doi:10.4172/2169-026X.1000173
- [7] Zheng, Z., Xie, S., Dai, H-N., Chen, X. and Wang, H. (2018) 'Blockchain challenges and opportunities: a survey', Int. J. Web and Grid Services, Vol. 14, No. 4, pp.352–375.
- [8] Daniel Karanja, Anthony Kiplang'at Sang, Mwangi Ndirangu. Influence of Integration of ICT on Human Resource Management in Kenyan Public Universities. International Journal of Sustainability Management and Information Technologies. Vol. 3, No. 6, 2017, pp. 73-78.doi: 10.11648/j.ijsm.20170306.13
- [9] B. A. Lloyd, "Professional networking on the internet," Conference Record of 2009 Annual Pulp and Paper Industry Technical Conference, 2009, pp. 62-66, doi: 10.1109/PAPCON.2009.5185427.
- [10] Ram Chhetri (2022) Impl ementation of blockchain technology into Human resource

recruitment Jyväskylä: JAMK University of Applied Sciences, February 2022, 63 pages.

[11] Sérgio Guerreiro, João F. Ferreira, Tiago Fonseca, Miguel Correia (2022) Integrating an Academic Management System with Blockchain: A case study, *Blockchain: Research and Applications*, 100099, ISSN 2096-7209, <https://doi.org/10.1016/j.bcra.2022.100099>.

[12] Ocheja, P., Flanagan, B., Ueda, H. et al. Managing lifelong learning records through blockchain. *RPTEL* 14, 4 (2019). <https://doi.org/10.1186/s41039-019-0097-0>

[13] Steve X. Chen\* (2017) *EchoLink - EKO – Blockchain Based Professional Networking and Recruiting Platform, Building A High-Trust Economy*. <https://github.com/EchoLinkTech/EchoLink>

[14] Grigore Albeanu (2017) Blockchain technology and education. *University, Bucharest, Romania*

[15] Sharples M., Domingue J. (2016): The Blockchain and Kudos: A Distributed System for Educational Record, Reputation and Reward. In: Verbert K., Sharples M., Klobučar T. (eds) *Adaptive and Adaptable Learning*. ECTEL 2016. Lecture Notes in Computer Science, vol 9891. Springer, Cham, [https://doi.org/10.1007/978-3-319-45153-4\\_48](https://doi.org/10.1007/978-3-319-45153-4_48).

## 7.2 Templates of data collection tools

### Recruitment Practices Questionnaire

This questionnaire is to evaluate the selection process of institutes. This information will be used in developing a system which aims to make the selection process fair and more effective.

h190305e@hit.ac.zw [Switch account](#)

Not shared

What channels do you usually use to advertise job openings?

Your answer

How effective do you think the above-mentioned channels are?

☐ Not Effective Enough

☐ Hardly Effective

☐ Effective

☐ Very Effective

What percentage of applicants are considered for interviews?

Your answer

On a scale of 1 to 10, how easy is it to tally all the applications?

1 2 3 4 5 6 7 8 9 10

Hard ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ Easy

What criteria do you use to compare applications and thus create a short list for interview eligible applicants?

Your answer

How effective is this method?

☐ Not Effective Enough

☐ Hardly Effective

☐ Effective

☐ Very Effective

How long does it usually take from the application deadline to get the shortlist of eligible applicants?

Your answer

What percentage of applicants would you say possess the minimum requirements for the job?

Your answer

How important is it to include academic results and transcripts in the application apart from mentioning them in the CV

1 2 3 4 5 6 7 8 9 10

Not important ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ Very important

### Recruitment Practices Questionnaire

h190305e@hit.ac.zw [Switch account](#)

Not shared

#### Verification of Information

Do you ever need to verify mentioned academic credentials?

☐ Yes

☐ No

☐ Sometimes

If yes, how do you do it and how long does the process usually take?

Your answer

What measures do have in place to protect against academic fraud?

Your answer

How do safeguard against bribery and favoritism in your selection process?

Your answer

[Back](#) [Submit](#) [Clear form](#)

Figure 22: Survey Questions



### 7.3 User manual of the working system

### 7.4 All users

#### 7.4.1 Log in

1. Visit the website homepage.
2. Click log in.
3. Enter your credentials into the MetaMask window that pops up.
4. Click log in.

### 7.5 Universities

#### 7.5.1 Adding a university

1. From the menu, click Add university.
2. Enter the details.
3. Click Save.

NB: The log in details will be sent to the university's email address.

#### 7.5.2 Adding a Company

1. From the menu, click Add Company
2. Enter details.
3. Click Save.

NB: The log in details will be sent to the company's email address.

#### 7.5.3 Adding A Course

1. From the menu, click Add Course
2. Enter details.
3. Click Save.

#### 7.5.4 Uploading Information

1. From the menu, click Upload <the data you want to upload>
2. Drag and drop the csv file with the information you want to upload.
3. Click Upload

NB: Please make sure the file columns match with those prescribed on the upload page.

## 7.6 Company

### 7.6.1 Creating a job opening

1. From the menu, click Add job opening.
2. Enter the details.
3. Click Save.

### 7.6.2 Adding job candidates

1. From the menu, click View Job openings.
2. Select the job opening you want.
3. From the top right corner, click View Possible Employees
4. Select a possible employee to view their profile.
5. Click select candidate.
6. The system returns to the possible employee's page. Repeat 4 and 5 until you have enough candidates for the job.

### 7.6.3 Sending out interview invitations

1. From the menu, click View Job openings.
2. Select the job opening you want.
3. From the top right corner, click Send Job Invitations

### 7.6.4 Add employee review.

1. From the menu, click manage employees.
2. For the employees whose review you want to add, click view profile.
3. navigate to the reviews part of the page.
4. enter review data.
5. click Save

## 7.7 Sample code

```
def AddCSVCourses(pathfile, uniaddress):
    result={}
    with open(pathfile, 'rb') as f:
        result = chardet.detect(f.read())
    read_data=pd.read_csv(pathfile, encoding=result['encoding'])
    datat_courses=['CourseName', 'CourseCode', 'CourseDescription',
'CourseDuration',
'CourseField', 'CourseLevel', 'CourseStartDate', 'CoursePrerequisites',
'CourseJobs']

    if set(datat_courses).issubset(read_data.columns):
        if TheUni.checkUni(uniaddress):
            for row in read_data.itertuples(index=True, name='Pandas'):
                ic=TheEdu.AddUniversityCourse(uniaddress,str(row.CourseName),str(
row.CourseCode).upper().replace("
",""),str(row.CourseDescription),str(row.CourseDuration),str(row.CourseField),str
(row.CourseLevel), "Active",str(row.CourseStartDate),"")
                if ic=="Success":
                    for a in str(row.CourseJobs).split("\n"):
                        ic1=TheEdu.AddUniversityCourseJob(uniaddress,str(row.Cour
seCode).upper().replace(" ", ""),a)
                        if str(row.CoursePrerequisites).split("\nOR\n")==[]:
                            pres=str(row.CoursePrerequisites).split("\n")
                        else:
                            pres=str(row.CoursePrerequisites).split("\nOR\n")
                        for a in pres:
                            ic2=TheEdu.AddUniversityCoursePrerequisites(uniaddress,st
r(row.CourseCode).upper().replace(" ", ""),a)
                            result[str(row.CourseName)]=ic
                        else:
                            result[str(row.CourseName)]= "Course already in the system."
                    return {'Status':'Success', 'Data':result}
            else:
                return {'Status':'msg', 'Data':"University Not Found"}
        else:
            return {'Status':'msg', 'Data':"CSV file has an invalid structure. Please
check the file and try again. Please check documentation to see the columns that
are required."}

#print(AddCSVCourses('C:\\Users\\Lenovo Yoga\\Desktop\\HIT 400 Pro
max\\flask\\csvs\\Courses.csv', "0xf200Dd274C6E061de66c658C26c22bB5596a71f2"))
```

```

def da_splitter(inp_txt):
    if inp_txt.find("\n")!=-1:
        return inp_txt.split("\n")
    elif inp_txt.lower().find(" or ")!=-1:
        return inp_txt.split(" or ")
    elif inp_txt.lower().find(" , ")!=-1:
        return inp_txt.split(" , ")
    else:
        return [inp_txt]

def AddCSVModules(pathfile, uniaddress):
    result={}
    with open(pathfile, 'rb') as f:
        result = chardet.detect(f.read())
    read_data=pd.read_csv(pathfile, encoding=result['encoding'])

    datat_modules=['ModuleCode', 'ModuleName', 'ModuleSkills', 'ModuleCredits',
        'ModuleDescription', 'ModuleStatus', 'ModuleStartDate', 'ModuleJobs',
        'Practical', 'ModuleCourses']
    if set(datat_modules).issubset(read_data.columns):
        if TheUni.checkUni(uniaddress):
            for row in read_data.itertuples(index=True, name='Pandas'):
                if str(row.Practical)=="Yes":
                    prac=True
                else:
                    prac=False
                ic=TheEdu.AddTheUniversityModule(uniaddress,str(row.ModuleName),s
tr(row.ModuleCode).upper().replace("
",""),str(row.ModuleDescription),str(row.ModuleCredits),"Active",str(row.ModuleSt
artDate),"",prac)
                if ic=="Success":
                    jobs=da_splitter(str(row.ModuleJobs))
                    for a in jobs:
                        ic1=TheEdu.AddUniModuleJob(uniaddress,str(row.ModuleCode)
.upper().replace(" ", ""),a)

                    pres=da_splitter(str(row.ModuleSkills))

                    for a in pres:
                        ic2=TheEdu.AddUniversityModuleSkill(uniaddress,str(row.Mo
duleCode).upper().replace(" ", ""),a)

                    crs=da_splitter(str(row.ModuleCourses))

```

```

        for a in crs:
            ic3=TheEdu.AddUniversityCourseModule(uniaddress,str(a).upper(),str(row.ModuleCode).upper().replace(" ", ""))

            result[str(row.ModuleName)]=ic
        else:
            result[str(row.ModuleName)]= "Module already in the system."

    return {'Status':'Success', 'Data':result}
else:
    return {'Status':'msg', 'Data':"University Not Found"}
else:
    return {'Status':'msg', 'Data':"CSV file has an invalid structure. Please check the file and try again. Please check documentation to see the columns that are required."}

#print(AddCSVModules('C:\\Users\\Lenovo Yoga\\Desktop\\HIT 400 Pro max\\flask\\csvs\\Modules.csv', '0xf200Dd274C6E061de66c658C26c22bB5596a71f2'))

def AddCSVStudents(pathfile, uniaddress):
    result={}
    with open(pathfile, 'rb') as f:
        result = chardet.detect(f.read())
    read_data=pd.read_csv(pathfile, encoding=result['encoding'])
    datat_students=['Name', 'Surname', 'NationalID', 'Email', 'Cell', 'DOB', 'Gender', 'Address', 'City', 'Part', 'CourseCode', 'CourseStatus', 'CourseStartDate', 'CourseEndDate']

    if set(datat_students).issubset(read_data.columns):
        if TheUni.checkUni(uniaddress):
            for row in read_data.itertuples(index=True, name='Pandas'):
                rez=Munhu.AddAUniStudent(str(row.Name),str(row.Surname),str(row.NationalID),str(row.Email),str(row.Cell),str(row.DOB),str(row.Gender),str(row.Address),str(row.City),str(row.Part),uniaddress)
                rez1=Munhu.AddAUniStudentCourse(str(rez['StudentAddress']),str(row.CourseCode).replace(" ", ""),str(row.CourseStatus),str(row.CourseStartDate),str(row.CourseEndDate),uniaddress)
                rez2=Munhu.AddAUniCourseStudent(str(row.CourseCode).upper().replace(" ", ""), str(rez['StudentAddress']), uniaddress)
                result[str(row.NationalID)]=rez
            else:
                result="University Not Found"

```

```

        return {'Status':'msg', 'Data':result}

    return {'Status':'Success', 'Data':result}
else:
    return {'Status':'msg', 'Data':"CSV file has an invalid structure. Please
check the file and try again. Please check documentation to see the columns that
are required."}

#print(AddCSVStudents('C:\\Users\\Lenovo Yoga\\Desktop\\HIT 400 Pro
max\\flask\\csvs\\Students.csv', '0xf200Dd274C6E061de66c658C26c22bB5596a71f2'))

def AddResults(pathfile, uniaddress):
    rez={}
    with open(pathfile, 'rb') as f:
        result = chardet.detect(f.read())
        read_data=pd.read_csv(pathfile, encoding=result['encoding'])
        datat_results=['NationalID', 'CourseCode', 'ModuleCode', 'Semester', 'Year',
'Grade',
        'Credits']

    if set(datat_results).issubset(read_data.columns):
        if TheUni.checkUni(uniaddress):
            for row in read_data.itertuples(index=True, name='Pandas'):
                _StudentAddress=Munhu.GetSTDAddress(row.NationalID)
                if _StudentAddress!="Student Not Found":
                    rezz=Munhu.AddAUniStudentModuleResult(_StudentAddress,
uniaddress, str(row.CourseCode).upper().replace(" ", ""), str(row.ModuleCode),
str(row.Semester), str(row.Year), str(row.Grade), str(row.Credits))
                    rez[str(row.NationalID)]=rezz
                else:
                    rez[str(row.NationalID)]=_StudentAddress

            return {'Status':'Success', 'Data':rez}
        else:
            return {'Status':'msg', 'Data':"University Not Found"}
    else:
        return {'Status':'msg', 'Data':"CSV file has an invalid structure. Please
check the file and try again. Please check documentation to see the columns that
are required."}

```

# Recruitment BloQs: A Blockchain Academic Record Based Recruitment System

Malvern Chatikobo; Linda Amos

*Department of Software Engineering, School of Information Sciences and*

*Technology Harare Institute of Technology, Harare, Zimbabwe*

[mchats55@gmail.com](mailto:mchats55@gmail.com); [lamos@hit.ac.zw](mailto:lamos@hit.ac.zw)

**Abstract**— The recruitment process for tertiary education students and graduates is plagued by academic fraud and inequality. To address these issues, we propose a blockchain-based platform that allows universities and tertiary institutions to post information about individuals' education, and companies to post job openings. The system matches candidates with the appropriate education for the job, ranks them based on academic excellence, and presents them to the employer. The platform is built on blockchain technology, making it reliable and secure. It also provides a fair and equal opportunity for students and graduates to showcase their skills and qualifications while reducing academic fraud. The proposed system offers a reliable, secure, and fair way to recruit tertiary education students and graduates, making the recruitment process more effective and accessible for companies.

**Keywords:** blockchain, recruitment, academic fraud, smart contracts, tertiary education.

## I. INTRODUCTION

The current recruitment process for tertiary education students and graduates is plagued with various issues such as academic fraud, unequal opportunities, and discrimination. As a result, many students struggle to find suitable employment, and companies struggle to identify the right candidates for the job. To address these issues, this paper proposes a blockchain-based platform that allows universities and other tertiary institutions to post information about individuals' education, and companies to add job opening information. The system then finds candidates with the appropriate education for the job, puts them in order of academic excellence, and shows them to the employer. After that, the system allows the employer to send out interview invitations with links candidates can use to RSVP for the

interview. All this information would be stored on a blockchain system, making it verifiable and valid. This paper discusses the proposed system, its features, and the benefits it offers to tertiary education students, graduates, and companies.

The use of blockchain technology in education and employment has been a subject of growing interest in recent years. The concept of using blockchain for education verification was first proposed by Mougayar in 2016 [1]. Since then, several studies have explored the potential of blockchain in the education sector. For instance, Thakur and Singh [2] proposed a blockchain-based platform for issuing digital degrees and certificates. They argued that the use of blockchain technology can help eliminate fraud and ensure the authenticity of academic records. Similarly, Kshetri [3] proposed a blockchain-based system for tracking and verifying the skills of job candidates. The system uses smart contracts to ensure the accuracy and transparency of the candidate's credentials.

In addition to addressing issues of academic fraud and verification, blockchain-based systems have the potential to promote equality and reduce discrimination in the recruitment process. For instance, Kim et al. [4] proposed a blockchain-based platform that uses smart contracts to ensure that job candidates are evaluated based on their skills and qualifications rather than their personal characteristics such as race, gender, or age. Similarly, Ong et al. [5] proposed a blockchain-based system that allows employers to access a diverse pool of job candidates and track their progress through the recruitment process.

Considering these developments, this paper proposes a blockchain-based platform that provides a reliable and secure way to store and verify the academic records of students and graduates. The platform will bring equality to the recruitment

process and reduce the amount of academic fraud, providing a fair and equal opportunity for students and graduates to showcase their skills and qualifications.

## II. PROBLEM STATEMENT

The recruitment process for tertiary education students and graduates is often marred by academic fraud and inequality. Past research has highlighted the prevalence of academic fraud in student records and resumes, which can significantly impact the hiring process [6][7]. Additionally, students from less privileged backgrounds may face discrimination and unequal opportunities, even if they possess the necessary qualifications and skills [8]. Students may falsify their academic records or misrepresent their qualifications, making it difficult for companies to identify the right candidates for the job. Similarly, students from less privileged backgrounds may face discrimination and unequal opportunities, even if they possess the necessary qualifications and skills.

To address these issues, this paper proposes a blockchain-based platform that provides a reliable and secure way to store and verify the academic records of students and graduates. The platform will bring equality to the recruitment process and reduce the amount of academic fraud, providing a fair and equal opportunity for students and graduates to showcase their skills and qualifications.

## III. RELATED WORK

QualiChain as described in [9] is a Portuguese blockchain based application that is used to store diplomas issued by Higher Education Institutions (HEIs). It is used to combat counterfeiting and falsification. It provides an interface for hiring companies to check the authenticity of an applicant's credentials. It is an extension of an already existing system called Fenix which is an academic management platform that supports full program and course management, including the issuing of diplomas (essentially PDF documents

that are printed, signed, and officially stamped).

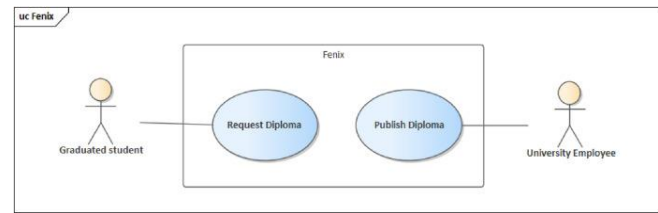


Figure 23: Use case definition of the Fenix

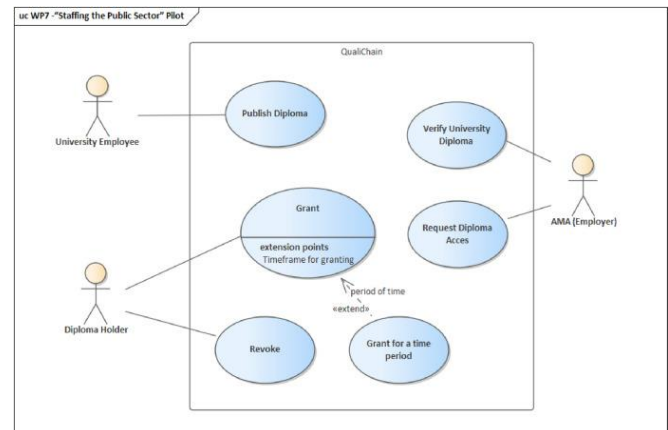


Figure 24: Use case definition of the QualiChain system

The blockchain of learning logs (BOLL) as proposed in [10] is a system that connects different institutions of learning and is used to transfer a student's records as they move between institutions. The system is used to keep track of a person's academic results and institutions attended in detail. It also solves the cold-start problem which is described as having a new academic record being created without being connected to the students' previous records.

The following 2 figures show how data is transferred between institutions outside the BOLL and on the BOLL respectively.



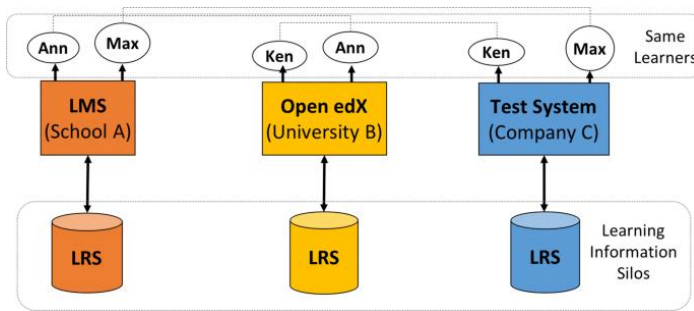


Figure 25: This shows an example of data movement outside the BOLL system.

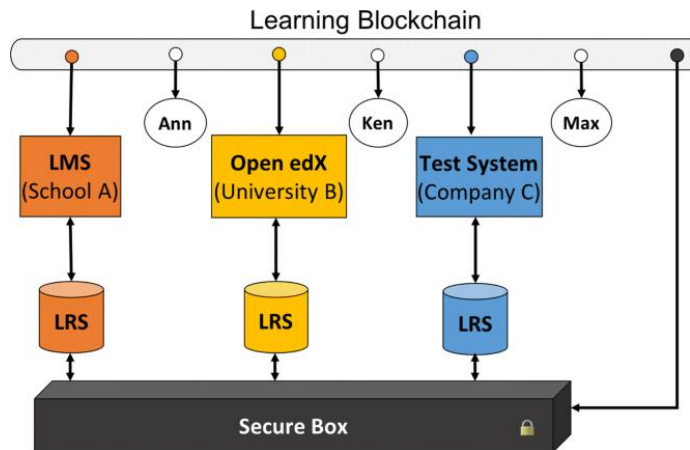


Figure 26: Proposed design of blockchain of learning logs (BOLL)

#### IV. SOLUTION

To eradicate the problems of the current recruitment processes, I aim develop a blockchain based academic record platform for selection in recruitment for use by companies to find employment eligible university students (current or past) with the following objectives:

- To develop a blockchain system to store relevant tertiary institute academic records and employment history records.
- To develop a system which can be used to verify any information pertaining to the skills and employment history of an individual.
- To develop a web platform for companies to find employment eligible university students (current or past)
- To develop a system that can map a student's proficient areas to an area of specialization for easier display to influence the decision making.

##### A. Features of the System

The proposed system will have the following features:

1. Blockchain-based platform: The system will be based on blockchain technology, which will make any information on it verifiable and valid. All academic records of

students and graduates will be stored on the blockchain, making it easy for companies to verify their qualifications and skills.

2. Academic record storage: The system will contain academic records of students eligible to be employed as interns or those who have graduated in different fields. These records will be stored on the blockchain, making any information on it verifiable and valid.
3. Company registration: Companies must register with any of the tertiary institutes and browse through viable candidates in the field of the position they want to fill. Initially, companies will have only access to a student's academic results and the skills they possess.
4. Smart contracts: Using smart contracts, students will be able to determine what the recruiters get to see. This feature will allow students to maintain control over their academic records and prevent any sensitive information from being disclosed to recruiters.

##### B. Benefits of the System

The proposed system offers the following benefits:

1. Fair and equal opportunity: The platform will bring equality to the recruitment process and reduce the amount of academic fraud, providing a fair and equal opportunity for students and graduates to showcase their skills and qualifications.
2. Reliable and secure: The blockchain-based platform provides a reliable and secure way to store and verify the academic records of students and graduates.
3. Easy and effective recruitment process: The system will make the recruitment process easier and more effective for companies, providing them with a pool of professionals to choose from based on their qualifications and academic results.

##### C. Solution Architecture

The client-side of the web app is implemented using HTML, CSS, and JavaScript. The user interacts with the client-side of the web app using a web browser. The client-side of the web app interacts with the server-side of the web app using HTTP requests.

The server-side of the web app is responsible for interacting with the Ethereum network and smart contracts. It is implemented using a backend Python flask as the backend. The server-side of the web app interacts with the smart contracts deployed on the Ethereum network using a web3.py library.

The web app uses the following architecture:

1. User Interface Layer: The user interface layer is the client-side of the web app. It consists of HTML, CSS, and JavaScript. The user interacts with the web app using a web browser. The user interface layer sends HTTP requests to the server-side of the web app.
2. Application Layer: The application layer is the server-side of the web app. It consists of a web server and an Ethereum client. The web server

receives HTTP requests from the user interface layer and processes them. The Ethereum client interacts with the Ethereum network and smart contracts.

3. Smart Contract Layer: The smart contract layer consists of the smart contracts deployed on the Ethereum network. The server-side of the web app interacts with the smart contract layer using the web3.py library.
4. Data Layer: The data layer consists of the data stored on the Ethereum network. The data is stored in a decentralized manner on the Ethereum network.

## V. RESULTS AND FUTURE WORKS

### A. Results



Figure 27: The interface of an individual's profile

The system performed well in allowing universities and tertiary institutions to post education information, companies to add job opening information, and finding candidates with the appropriate education for the job and the results are summarized in Table I.

Table 9: OBJECTIVE ACHIEVED

Objectives	Fully achieved	Partially achieved
To develop a blockchain system to store relevant tertiary institute academic records and employment history records	✓	
To develop a system which can be used to verify any information pertaining skills and employment history of an individual	✓	

To develop a web platform for companies to find employment eligible university students (current or past)	✓	
To develop a system that can map a student's proficient areas to an area of specialization for easier display to influence the decision making	✓	
To develop a system which can be used to verify any information pertaining skills and employment history of an individual	✓	
To develop a web platform for companies to find employment eligible university students (current or past)	✓	

### B. Future Works

Some potential areas of future work for the system could include:

1. Expanding the system to include additional types of educational institutions, such as vocational schools or certification programs.
2. Integrating with HR software to provide a seamless hiring process for employers.
3. Adding machine learning algorithms to improve the matching process between candidates and job openings.
4. Incorporating more advanced security measures for the blockchain storage system, such as multi-factor authentication and encryption.

## VI. CONCLUSION

In conclusion, the proposed blockchain-based platform for the recruitment process of tertiary education students and graduates offers a reliable, secure, and fair way for students and graduates to showcase their skills and qualifications to companies. The platform provides a secure way to store and verify academic records, reducing the amount of academic fraud, and bringing equality to the recruitment process. The proposed system will provide a fair and equal opportunity for students and graduates to showcase their skills and qualifications and make the recruitment process easier and more effective for companies.

## ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere gratitude to all those who have contributed to the successful completion of this project.

First and foremost, I would like to thank God, whose grace has been sufficient from that first day of pre-school to this milestone I'm on the verge of achieving.

I am thankful to my supervisor, Ms. L. Amos, for her invaluable guidance, expert advice, and unwavering support throughout the project. Her mentorship has been instrumental in shaping the direction of this work and has helped me to develop important skills and knowledge in the field.

Furthermore, I would like to express my gratitude to the staff of the Software Engineering department for their assistance and support, especially during the challenging times of the project. The department's resources and facilities have been crucial in carrying out the research and development tasks.

I would like to thank my mother and aunt for being warriors and fighting for me and my siblings to have a quality education. I would also like to thank the rest of my family and friends for their unwavering support, encouragement, and love. Their constant belief in my abilities has been a driving force in overcoming challenges and achieving success.

Finally, I would like to thank all the participants who volunteered their time and efforts to provide valuable data for this project. Their cooperation and contribution have been instrumental in the successful completion of this work.

Once again, I extend my sincere appreciation to everyone who has played a role in this project. Your support and contributions have been invaluable, and I am deeply grateful for your presence in my life.

#### REFERENCES

- [1] W. Mougayar, "The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology", John Wiley & Sons, 2016.
- [2] S. Thakur and P. Singh, "A blockchain based approach for digital degree issuance system", *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 5, pp. 663-666, 2019.
- [3] N. Kshetri, "Blockchain's roles in meeting key supply chain management objectives", *International Journal of Information Management*, vol. 39, pp. 80-89, 2018.
- [4] M. Kim, H. Park, and S. H. Kim, "Blockchain-based secure and transparent employment service", *IEEE Access*, vol. 6, pp. 21853-21863, 2018.
- [5] L. Ong, C. S. Poon, and P. Zhang, "Blockchain-based e-recruitment system", in *Proceedings of the 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 587-591, 2018.
- [6] D. McCabe, "Cheating among college and university students: A North American perspective," *International Journal for Educational Integrity*, vol. 1, no. 1, pp. 1-14, 2005.
- [7] M. G. LaForge and A. L. Vitale, "Academic Fraud and Employment Screening," *Journal of Business and Psychology*, vol. 22, no. 2, pp. 137-145, 2007.
- [8] D. Pager and B. Western, "Race at work: Realities of race and criminal record in the employment process," *National Bureau of Economic Research, Working Paper No. 13216*, 2007.
- [9] Sérgio Guerreiro, João F. Ferreira, Tiago Fonseca, Miguel Correia (2022) Integrating an Academic Management System with Blockchain: A case study, *Blockchain: Research and Applications*, 100099, ISSN 2096-7209, <https://doi.org/10.1016/j.bcra.2022.100099>.
- [10] Ocheja, P., Flanagan, B., Ueda, H. et al. Managing lifelong learning records through blockchain. *RPTEL* 14, 4 (2019). <https://doi.org/10.1186/s41039-019-0097-0>