Discovering Knowledge

FINAL YEAR PROJECT REPORT
2022

Degree Verification System Using Blockchain

Group Members

Student Name	Enrolment#
Adil Waheed	02-131192-082
Sumayya Khalid	02-131192-057
Rabya Essani	02-131192-081

Supervised by

Sir Faisal

Department of Software Engineering
BAHRIA UNIVERSITY KARACHI CAMPUS

Intellectual Property & Submission Policy

This is to declare that,

1) The project under the supervision of **Sir Faisal** having title **Degree Verification System Using Blockchain** carried out in partial fulfillment of the Bachelor of Engineering in Software Engineering degree program requirements and is the sole property of the Bahria University.
2) This report is submitted as the requirement for the project in accordance with the rules laid down by the Bahria University as part of the requirements for the award of the degree of Bachelor of Engineering in Software Engineering.
3) The work presented in this report is our own except where due reference or acknowledgment given to the work of others.
4) We are aware that Bahira University asserts legal and beneficial ownership rights over all Intellectual Property developed as a result of support ly from or channeled through Bahria University.
5) We are agreed to assign to Bahira University all of their rights, title and interest in and to Intellectual Property developed as a result of utilization of Bahira University Resources including copyright in any material that is teaching material, computer programs, or created at the request or direction of Bahira University.

S.No.	Student Name	Enrolment #	Signatures
1	Adil Waheed	02-131192-082	___________
2	Sumayya Khalid	02-131192-057	___________
3	Rabya Essani	02-131192-081	___________

Reference:

[1] R&D policy handbook (BUORIC-P15)

Submission Proforma

It is acknowledged that I as a supervisor gone through this report. The report contains all the essential sections as required by the department in accordance with the rules laid down by the Bahria University.

Supervisor Name: Sir Faisal

Supervisor Signatures: ______________________

Dated: ______________________

Acknowledgments

We would want to express our gratitude to everyone who helped us complete this project successfully. We would like to thank **Sir Faisal**, our research supervisor, for her important advice, guidance, and immense patience throughout the research's growth. We acknowledge and appreciate the Head of department who gave us this golden opportunity to develop this project that helped us in doing a lot of research due to which we came to know about so many new things for which we are really thankful to him.

We'd also like to thank our supportive parents and friends for their assistance and encouragement.

Abstract

In Pakistan, there are around 37 million students who graduate each year. Some of these students choose to pursue higher education abroad while others enter the workforce. One issue that arises in this process is the falsification of documents. Verifying documents can be a difficult and time-consuming process. The use of blockchain technology offers a solution to this problem by creating a decentralized system for verifying academic degrees.

We propose a Decentralized certificate verification system based on blockchain technology and identify the security considerations that need to be addressed in such a system. Currently, manual verification of degree certificates by third parties during the admission and interview process takes a lot of time and resources. The proposed digital certificate system based on blockchain technology aims to address the issue of certificate forgery.

Keyword: Blockchain, Decentralized, IPFS (Interplanetary File System), Certificate Verification and Generation, E-Certificate, Ethereum, React.js, Smart Contract, Solidity

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Introduction

Each year, there are many students graduating in our country, making it challenging to track and verify such a large number of records. This can lead to the production of fake or duplicate certificates and the tampering of records. There are many agencies operating secretly in our country that are involved in this scam so, it can be difficult and time-consuming to distinguish between a fake and an original certificate.

Blockchain technology offers a solution to these issues. Data in a blockchain cannot be changed under normal circumstances and any tampering can be quickly detected, data or nodes are only validated when multiple parties approve them, making the system reliable and authenticated at all times. The system we propose will not only validate certificates but also generate them, saving time and eliminating the need for students to worry about losing or damaging physical certificates.

1.2 Problem Statement

- One issue with the current degree verification process is that a student must have their certificates at each stage of validation, which increases the risk of losing or damaging them.
- It can also be time-consuming for the person responsible for validation to authenticate each degree.
- There are many agencies operating secretly in our country that are involved in creating the fake degrees of Student so, it can be difficult and time-consuming to distinguish between a fake and an original certificate.

1.3 Background / Literature Review

There are many research paper on internet related to degree verification process but in each paper there is only one method available to verify degree, But when we are talking about Pakistan there is no system who verify degree online. Currently, degrees are issued to students as physical copies and there is no digital way to verify these degree. There is also no platform to safely store and verify the certificates when needed. This can lead to the creation of fake graduation degrees in order to obtain unauthorized jobs. In the workplace, HR teams or third parties manually verify the educational background of employees, which can be time-consuming and subject to delays. It can also be difficult to distinguish between fake and original degrees if the master register has been tampered with. Some universities do store certificates in digital form, but these are often on centralized networks where there is a risk of tampering. As a result, there is a lack of security and integrity in both manual and digital systems, which can increase the likelihood of fraud.

1.4 Problem in existing system

One issue with the current degree verification process is that a student must have their certificates at each stage of validation, which increases the risk of losing or damaging them. It can also be time-consuming for the person responsible for validation to authenticate each degree. There are many agencies operating secretly in our country that are involved in creating the fake degrees of Student so, it can be difficult and time-consuming to distinguish between a fake and an original certificate.

1.5 Proposed Solution

Blockchain technology offers a solution to these issues. Data in a blockchain cannot be changed under normal circumstances and any tampering can be quickly detected, data or nodes are only validated when multiple parties approve them, making the system reliable and authenticated at all times. The system we propose will not only validate certificates but also generate them, saving time and eliminating the need for students to worry about losing or damaging physical certificates.

1.5.1 Features of the project

FE-1: It is not possible for anyone to alter or create fake degrees using this system.
FE-2: The verification process for employers is made easier and more efficient using this System.
FE-3: Only authorized admin is able to add marks to the blockchain.
FE-4: Student also have E-Degree and they don’t need to worry about losing their degree.
FE-5: Verifier check the authenticity of degree by using 3 ways.

1-Upload document
2-Enter Unique id of degree
3-Scan QR-Code

1.5.2 Methodology/Algorithm

- The student’s degree/transcript and details will be uploaded by the Institute.
- Using these details, a pdf of the degree/transcript is Generated
- The pdf is hashed
- The hash value generated in the above step is stored on the blockchain after hashing it with the private key
- The degree/transcript is mailed to the student.

1.5.3 Technologies to be used

You may write specification regarding hardware, operating system, development framework, and libraries.

- **React:** Front-End Programming Language & Framework
- **NodeJS**: Backend Language
- **Ethereum**: Decentralized open-source Blockchain
- **MongoDB**: Type of NoSQL Database
- **Solidity:** Solidity is a programming language used to write smart contracts for the Ethereum blockchain.
- **Smart Contract**: Language similar to TypeScript
- **Compiler and IDE:** VS Code

1.6 Project Scope / Deliverables

The main aim of using blockchain technology for online degree verification and generation is to provide a secure and tamper-proof way to store and verify academic degrees. The decentralized nature of the blockchain means that records are stored on multiple servers and can only be altered with the consensus of the network, making it difficult for anyone to forge or alter degree records without being detected. This can help to prevent fraud and improve the reliability of academic records. Additionally, using blockchain technology can make the verification process more efficient and accessible, as records can be easily accessed and verified online.

1-Admin Module:

Admin is login into the system and is responsible to enter the details of Student and generate the certificate with unique ID and QR-code.

2-Verifier Module:

When a student applies for Interview or Higher Education, the validator, who is an authority at the organization, must verify the authenticity of the student's documents. The validator has three options for doing this:

1. Uploading the documents
2. Entering the student's ID.
3. Scan QR-code

2. SOFTWARE PROJECT MANAGEMENT PLAN

2.1 Project Organization

Keeping project organized and following a software process model provides structure when designing and building of applications. A framework that outlines the tasks that need to be performed in each phase of software development. These steps allow developers to analyze the requirements. For organizing our Project, we will be using Agile Methodology. We use Kanban Approach in Our FYP. In the context of agile software development, Kanban can be used to visualize and manage the flow of work through the development process.

2.2 Software Process Model

In this project, we will be using the agile development methodology and specifically the Kanban approach to manage and track our progress. We will use visual aids, such as a Kanban board, to understand the status of the project and identify any issues or bottlenecks. We will also use Jira software to depict our progress and to collaborate with team members. In addition to the Kanban board, we will also use cumulative frequency diagrams to help us understand the project's situation and make informed decisions.

One of the key benefits of using Kanban is that it allows us to be flexible and responsive to changes. We can deliver increments of the project at any time and are always open to new ideas or modifications from the client. The Kanban board will help us to visualize the different stages of the project, such as designing, coding, testing, and deployment, and we will set work-in-progress (WIP) limits to ensure that the workflow remains smooth.

Some of the major milestones for our project include front-end and back-end development. The number of versions we create will depend on the proposed changes requested by the client after the initial deployment. By using Kanban and other agile techniques, we hope to deliver high-quality results efficiently and effectively.

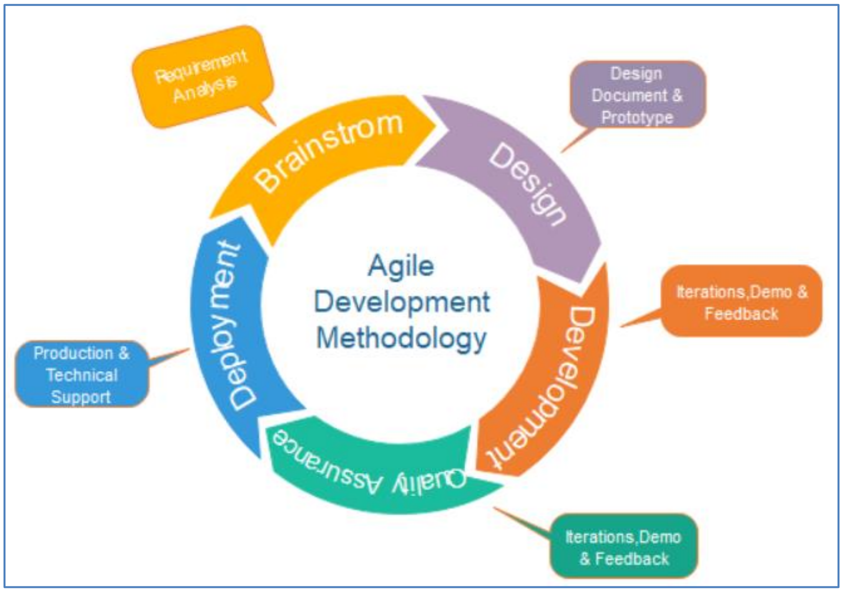
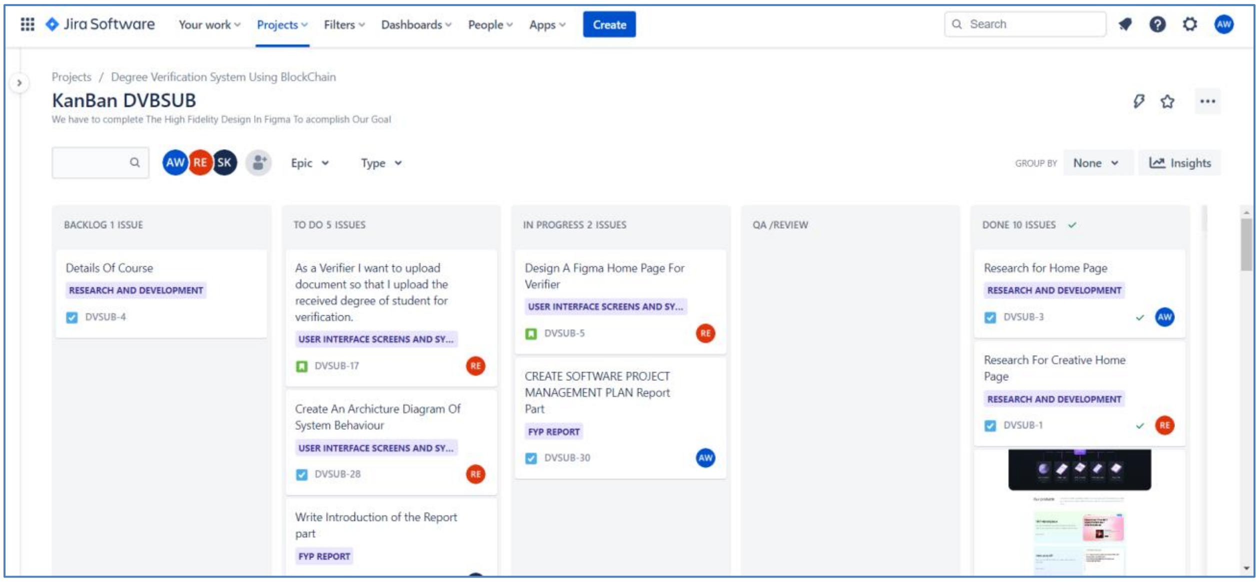

Requirement Analysis
Design Document & Prototype
Brainstrom
Design
Agile Development Methodology
Deployment
Development
Iterations,Demo & Feedback
Production & Technical Support
Quality Assurance
Iterations,Demo & Feedback

Figure1: Agile Development



Projects / Degree Verification System Using BlockChain

KanBan DVBSUB

We have to complete The High Fidelity Design In Figma To acomplish Our Goal

BACKLOG 1 ISSUE

- Details Of Course — RESEARCH AND DEVELOPMENT — DVSUB-4

TO DO 5 ISSUES

- As a Verifier I want to upload document so that I upload the received degree of student for verification. — USER INTERFACE SCREENS AND SY... — DVSUB-17 — RE
- Create An Archicture Diagram Of System Behaviour — USER INTERFACE SCREENS AND SY... — DVSUB-28 — RE
- Write Introduction of the Report part — FYP REPORT

IN PROGRESS 2 ISSUES

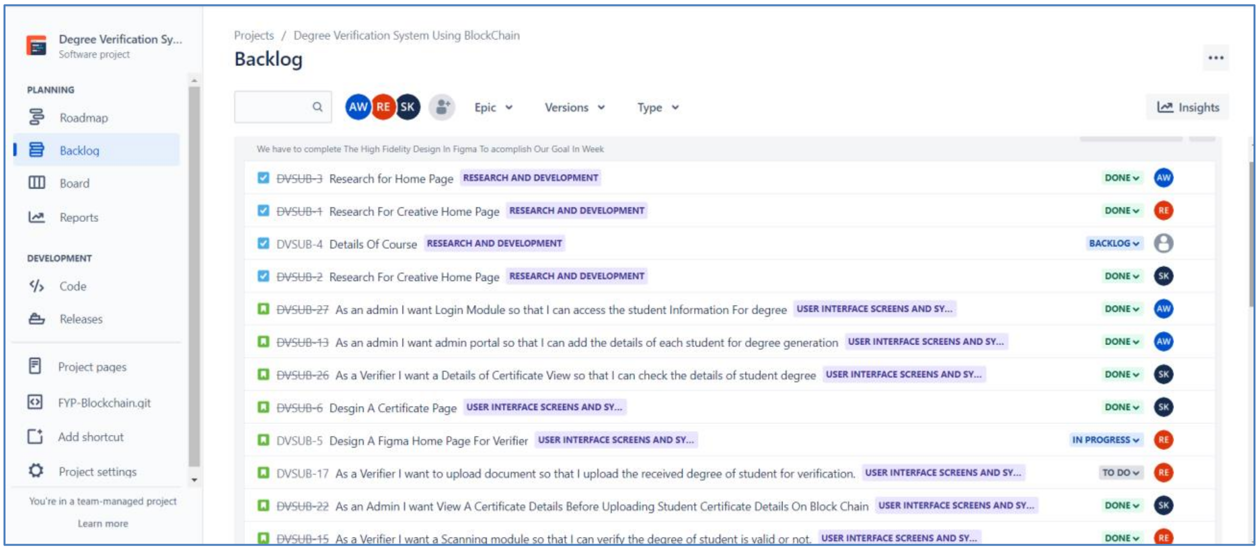
- Design A Figma Home Page For Verifier — USER INTERFACE SCREENS AND SY... — DVSUB-5 — RE
- CREATE SOFTWARE PROJECT MANAGEMENT PLAN Report Part — FYP REPORT — DVSUB-30 — AW

QA /REVIEW

DONE 10 ISSUES ✓

- Research for Home Page — RESEARCH AND DEVELOPMENT — DVSUB-3 ✓ — AW
- Research For Creative Home Page — RESEARCH AND DEVELOPMENT — DVSUB-1 ✓ — RE

Figure 2: Kanban Board of Project



Projects / Degree Verification System Using BlockChain

Backlog

We have to complete The High Fidelity Design In Figma To acomplish Our Goal In Week

~~DVSUB-3~~	Research for Home Page	RESEARCH AND DEVELOPMENT	DONE	AW
~~DVSUB-1~~	Research For Creative Home Page	RESEARCH AND DEVELOPMENT	DONE	RE
DVSUB-4	Details Of Course	RESEARCH AND DEVELOPMENT	BACKLOG	
~~DVSUB-2~~	Research For Creative Home Page	RESEARCH AND DEVELOPMENT	DONE	SK
~~DVSUB-27~~	As an admin I want Login Module so that I can access the student Information For degree	USER INTERFACE SCREENS AND SY...	DONE	AW
~~DVSUB-13~~	As an admin I want admin portal so that I can add the details of each student for degree generation	USER INTERFACE SCREENS AND SY...	DONE	AW
~~DVSUB-26~~	As a Verifier I want a Details of Certificate View so that I can check the details of student degree	USER INTERFACE SCREENS AND SY...	DONE	SK
~~DVSUB-6~~	Desgin A Certificate Page	USER INTERFACE SCREENS AND SY...	DONE	SK
DVSUB-5	Design A Figma Home Page For Verifier	USER INTERFACE SCREENS AND SY...	IN PROGRESS	RE
DVSUB-17	As a Verifier I want to upload document so that I upload the received degree of student for verification.	USER INTERFACE SCREENS AND SY...	TO DO	RE
~~DVSUB-22~~	As an Admin I want View A Certificate Details Before Uploading Student Certificate Details On Block Chain	USER INTERFACE SCREENS AND SY...	DONE	SK
~~DVSUB-15~~	As a Verifier I want a Scanning module so that I can verify the degree of student is valid or not.	USER INTERFACE SCREENS AND SY...	DONE	RE

Figure 3: Backlog of Project

2.3 Roles and Responsibilities

DVSUB project is assigned to three members Adil Waheed, Sumayya Khalid, Rabya Essani. Three members are responsible for whole development of software. We will use GitHub to store our code and use versioning tool to maintain different versions of software and we Use Jira Software Tool for Monitoring and Controlling the Progress of our Project. All the documentation will be uploaded on Jira Tickets as well as google drive so that each member can access them anywhere and anytime.

Team Members	Role
Adil Waheed	Creating Backlog, Designer, Back-End Developer, Documentation, Monitoring
Sumayya Khalid	Designer, Front-End and Back-End Developer, Documentation
Rabya Essani	Designer, Back-End Developer, Tester, Documentation

2.4 Tools and Techniques

2.4.1 Front-End Tool:

- HTML, CSS & JavaScript.
- Media Queries for responsiveness.
- Bootstrap for responsiveness.
- React.js Library
- Figma For Designing (Prototype)

2.4.2 Back-End Tool:

- Mongo DB for Storing User Session
- Node.js

2.4.3 Other Languages and Testing Tool

- Solidity language for Smart Contract
- Ganache cli for local testing
- Truffle Command Line Tool (Development Environment Tool)
- EVM (Ethereum Virtual Machine)
- GitHub For version Control
- Jira for keeping track of project
- Rinkeby

2.5 Project Management Plan

2.5.1 Tasks

There are some tasks involved in executing phase of projects. Some tasks are shown below:

- Requirement elicitation & analysis/Write SRS.
- Designing the system Prototype to get Feedback from Client. / Write the SDD.
- Implementation Version 1/ Development Version 1.

2.5.2 Task-1

Create Software Requirements Specifications (SRS) DVSUB-32

2.5.2.1 Description

Requirements will be gathered from client where he will tell us about functional & non-functional requirements. Our team should be aware of what we are going to build and for whom, so that project can be developed as per clients’ expectations and yields best results.

2.5.2.2 Deliverables and Milestones

- Software Requirement Specification (SRS)
- Analysis Document

2.5.2.3 Resources Needed

- Paper and Pen.
- Existing software to get ideas.
- Laptop for saving information.
- Good Communication Skills.

2.5.2.4 Dependencies and Constraints

Client must visit existing websites to get an idea of how his product is going to look like after development. Client should be aware of some technical terms like gas fees, transaction, Wallet address etc.

2.5.2.5 Risks and Contingencies

If our group member will not understand the newly technology of blockchain and its concept of contracts so will find another way to complete our project as soon as possible.

2.5.3 Task-2

Create Software Design Description (SDD) DVSUB-33

2.5.3.1 Description

Now we will create a virtual architecture of whole system where we will depict number of modules, interfaces, API, database, and their relationships so that developer can get a clear picture of what he/she is going to build.

2.5.3.2 Deliverables and Milestones

- Software Design Description (SDD)
- Number Of Modules
- Use Cases Diagram
- Sequence Diagram
- System Architecture Design.
- Work Flow Diagram
- Programming Languages to be used

2.5.3.3 Resources Needed

- Software Requirement Specification (SRS)
- Project team Collaboration
- Laptop for creating document
- Ms. Visio Software
- Figma For Design A Prototype
- Paper and Pencil to design document.

2.5.3.4 Dependencies and Constraints

Complete SRS is required altogether with product owner who represents client so that accurate SDD can be created. Experienced developers are required so that accurate SDD can be created.

2.5.3.5 Risks and Contingencies

Change of requirements after requirement elicitation can lead to increase in time and cost as it disturbs the scope in triangle. Secondly what client wants are not properly written in a form of requirement as the person gathering requirement is unable to create clear picture.

2.5.4 Task-3

Implementation DVSUB-45

2.5.4.1 Description

Developers will develop the designed software with the decided language. SDD will provide to developers while implementation. As we are following agile methodology hence continuous feedback will be taken from client.

2.5.4.2 Deliverables and Milestones

- Developed modules.
- Unit Testing report
- Debugging reports
- Status Reports

2.5.4.3 Resources Needed

- Software Design Document (SDD)
- Laptop/ desktop for programming
- Communication methods.
- Vs Code
- Ganache
- EVM
- Meta Mask wallet
- Rinkeby

2.5.4.4 Dependencies and Constraints

Programming skills are required to complete the task under cost and time constraints. Accurate and unambiguous software design document are required to develop software as per client expectations.

2.5.4.5 Risks and Contingencies

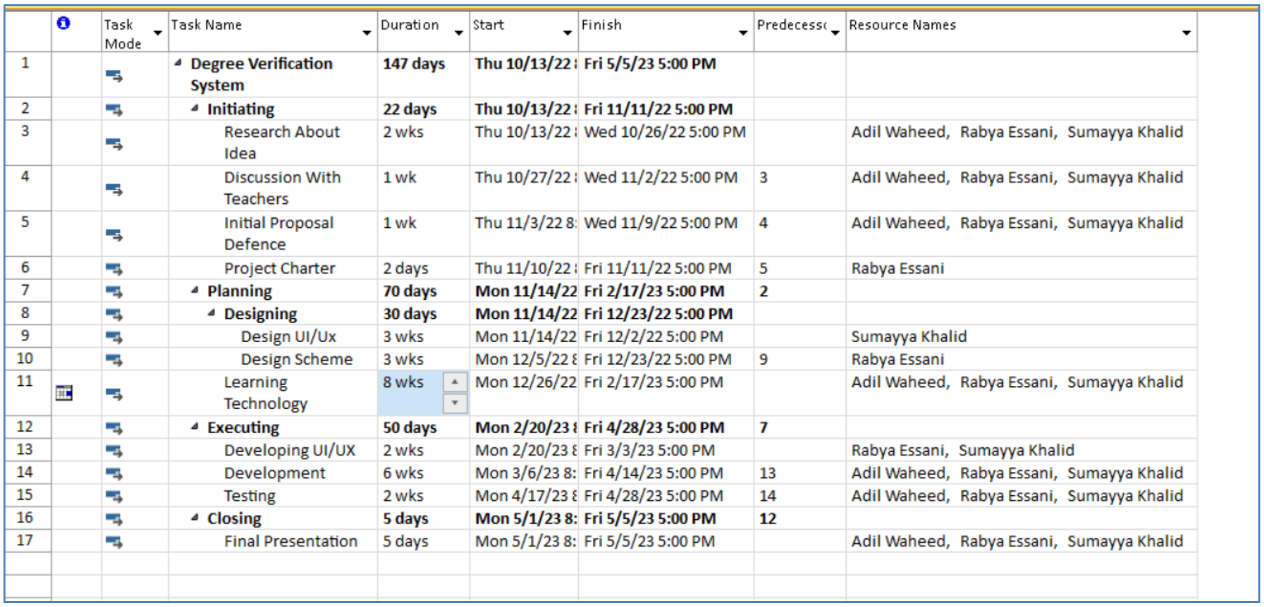
Vague SDD can lead to undesirable results and client will be unsatisfied. Secondly lack of skill can increase delivery time and increases cost.

2.6 Assignments

For group projects, identify the assignment of team members to tasks.

Tasks	Performed By
Information gathering related to similar projects	Adil Waheed Rabya Essani Sumayya Khalid
Defining Scope	Sumayya Khalid Rabya Essani
Communicator	Adil Waheed Rabya Essani Sumayya Khalid
Software Requirement Specification	Rabya Essani
Software Project Management Plan	Adil Waheed
Software Design Document	Adil Waheed Rabya Essani Sumayya Khalid
Front-End Development	Rabya Essani Sumayya Khalid Sumayya Khalid
Back-End Development	Adil Waheed Rabya Essani Sumayya Khalid
Use Case Diagram	Rabya Essani
Sequence Diagram	Rabya Essani
Work Breakdown Diagram	Adil Waheed
Workflow Diagram	Sumayya Khalid
Architectural Diagram	Rabya Essani
Unit Testing	Adil Waheed Sumayya Khalid
User Acceptance Testing	Rabya Essani Sumayya Khalid
Report Formation	Sumayya Khalid
Quality Assurance	Adil Waheed Rabya Essani Sumayya Khalid
Research & Development	Adil Waheed Rabya Essani Sumayya Khalid

2.7 Timetable



		Task Mode	Task Name	Duration	Start	Finish	Predecess	Resource Names
1			**Degree Verification System**	**147 days**	**Thu 10/13/22 8**	**Fri 5/5/23 5:00 PM**		
2			**Initiating**	**22 days**	**Thu 10/13/22 8**	**Fri 11/11/22 5:00 PM**		
3			Research About Idea	2 wks	Thu 10/13/22 8	Wed 10/26/22 5:00 PM		Adil Waheed, Rabya Essani, Sumayya Khalid
4			Discussion With Teachers	1 wk	Thu 10/27/22 8	Wed 11/2/22 5:00 PM	3	Adil Waheed, Rabya Essani, Sumayya Khalid
5			Initial Proposal Defence	1 wk	Thu 11/3/22 8:	Wed 11/9/22 5:00 PM	4	Adil Waheed, Rabya Essani, Sumayya Khalid
6			Project Charter	2 days	Thu 11/10/22 8	Fri 11/11/22 5:00 PM	5	Rabya Essani
7			**Planning**	**70 days**	**Mon 11/14/22**	**Fri 2/17/23 5:00 PM**	**2**	
8			**Designing**	**30 days**	**Mon 11/14/22**	**Fri 12/23/22 5:00 PM**		
9			Design UI/Ux	3 wks	Mon 11/14/22	Fri 12/2/22 5:00 PM		Sumayya Khalid
10			Design Scheme	3 wks	Mon 12/5/22 8	Fri 12/23/22 5:00 PM	9	Rabya Essani
11			Learning Technology	8 wks	Mon 12/26/22	Fri 2/17/23 5:00 PM		Adil Waheed, Rabya Essani, Sumayya Khalid
12			**Executing**	**50 days**	**Mon 2/20/23 8**	**Fri 4/28/23 5:00 PM**	**7**	
13			Developing UI/UX	2 wks	Mon 2/20/23 8	Fri 3/3/23 5:00 PM		Rabya Essani, Sumayya Khalid
14			Development	6 wks	Mon 3/6/23 8:	Fri 4/14/23 5:00 PM	13	Adil Waheed, Rabya Essani, Sumayya Khalid
15			Testing	2 wks	Mon 4/17/23 8	Fri 4/28/23 5:00 PM	14	Adil Waheed, Rabya Essani, Sumayya Khalid
16			**Closing**	**5 days**	**Mon 5/1/23 8:**	**Fri 5/5/23 5:00 PM**	**12**	
17			Final Presentation	5 days	Mon 5/1/23 8:	Fri 5/5/23 5:00 PM		Adil Waheed, Rabya Essani, Sumayya Khalid

Figure 4: Table

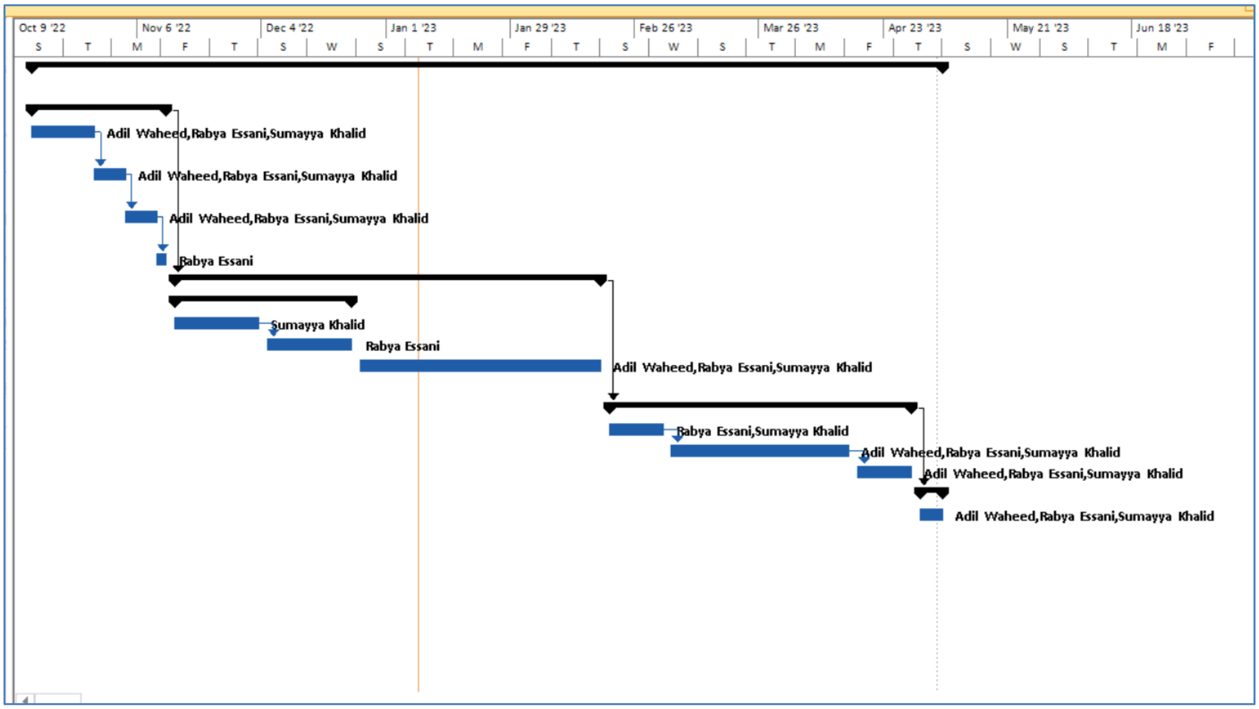


Figure 5: Gantt Chart

3. SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 Introduction

A degree verification system based on blockchain technology is a decentralized and secure system that uses blockchain technology to store and verify degree information. It allows students, alumni, and employers to verify degree information in a fast and secure way.

3.2 Product Overview

A blockchain-based degree verification system is a system that uses blockchain technology to securely and immutably store and verify educational credentials such as degrees. The system allows educational institutions to issue and verify degrees and transcripts on the blockchain, where they can be easily accessed and verified by employers and other organizations. The use of blockchain technology ensures that the credentials are secure and tamper-proof, and that the verifier can trust the authenticity of the information. Additionally, the decentralized and distributed nature of blockchain can allow for interoperability between institutions, making the verification process more efficient and streamlined.

3.3 Specific Requirements

1. Institution is able to issue degrees on Web application.
2. Web application can generate pdf with QR-code and Unique ID attached on document uploaded by institution.
3. Web application able to send that document to IPFS to create a hash.
4. IPFS returns a hash of document to system, further system can store hash and Unique ID on blockchain.
5. System can mail E-document to student.
6. Students can access and share their credentials with potential employers and other organizations.
7. Organization can verify documents in either of the 3 ways mentioned below:
   - Upload Document pdf
   - Scan QR-code
   - Enter Unique ID
8. Website should be available 24/7.

3.4 Functional and Data Requirements

- **Secure and tamper-proof storage of degree information: Essential**
  Blockchain technology uses cryptography to secure the data and ensures that once data is recorded on the blockchain it cannot be altered, this ensures the secure and tamper-proof storage of degree information.

  - **Input**
    Institute will upload students' data such as Student's name, degree program, Passing year, and CGPA on portal.

  - **Output**
    The system will create an E-certificate with a Unique ID and QR code on runtime.

  - **Changes in System**
    E-certificate of each student shown to the institute.
  - **Functional Requirements**
    Students' data is stored on the blockchain which can no longer be altered.

- **Verification and Validation:**
  The system must be able to verify and validate the authenticity of degree information and the identity of the person presenting it.
  - **Input**
    QR-Code, Document or Unique ID
  - **Output**
    Valid or Invalid Document retrieved
  - **Changes in System**
    E-certificate of each student shown if available
  - **Functional Requirements**
    Students' data is stored on blockchain which can no longer be altered.

- **Encryption and digital signature:**
  The system must be able to encrypt and digitally sign the degree information before it is added to the blockchain, in order to ensure the security of the data.
  - **Input**
    QR-Code, Document or Unique ID
  - **Output**
    E-degree.
  - **Changes in System**
    E-degree successfully stored in blockchain
  - **Functional Requirements**
    Encrypted Data

- **Searching and retrieving:**
  The system must allow to search and retrieve specific degree information and provide data in a format that can be easily understood by end users.
  - **Input**
    Student Data
  - **Output**

QR-code and E-degree.

- **Changes in System**
  E-degree with QR-code send to university

- **Functional Requirements**
  Specific degree information retrieved.

- **Smart contract execution:**
  The system must be able to execute smart contracts that are used to automate the verification process and ensure that all requirements are met before the degree is verified and added to the blockchain.

3.5 Non-Functional Requirements

3.5.1 Reliability

The System needs to support that data audited publicly does not leak private information such as hash-code of blockchain.

3.5.2 Availability

Web should be available 24/7. Server should not crash when there is a lot of traffic.

3.5.3 Security

One important aspect of security in a blockchain-based degree verification system is that we will use of digital signatures and encryption to secure the data stored on the blockchain. Digital signatures are used to ensure that only authorized parties can add or modify data on the blockchain, while encryption is used to protect the data from unauthorized access.

Another important aspect is that we will use private and public keys to secure access to the data on the blockchain. In this system, individual students would be issued a unique private key that would be used to prove their identity and access their data on the blockchain.

3.5.4 Maintainability

We will follow iterative enhancement model for maintenance as we will keep our software operational while performing maintenance. Maintenance can of any time like corrective, adaptive, perfective and preventive depending upon the situation. As our application will be built on three tier architecture, therefore maintaining one layer will have no impact on other layer.

3.5.5 Portability

As this is web-based application hence it can be accessed from anywhere around the world and at any time, there are no constraints of operating system, browser and etc. Our application will be responsive and will work on Opera, Chrome, Safari, Microsoft Edge and etc.

3.5.6 Performance

User should have

- Meta mask: wallet
- high-speed internet at-least 20Mbps
- Updated browser like Chrome of version 96.0.4664.110
- Scalability of website is up-to 500 users.
- Minimum Storage to save PDF file 10GB.

3.5.7 Usability Requirements

We will make our website responsive through media queries used in CSS. Our website will work on all available screen size. We will use heading and sub-headings so that information can be more understandable to the user. We will reduce page loading time by reducing file sizes and optimizing CSS files. We maintain consistency across the website by using similar fonts and colors.

3.6 Proposed Solution

3.6.1 Features of Project:

1. Decentralization: The use of a decentralized blockchain network will ensure that the system is not controlled by a single entity and can be audited by anyone.

2. Tamper-resistant data: Data stored on the blockchain will be protected by cryptographic techniques, making it tamper-resistant and resistant to hacking.

3. Digital signature and encryption: Digital signatures and encryption will be used to secure the data stored on the blockchain and to ensure that only authorized parties can add or modify data on the blockchain.

4. Smart contracts: Smart contracts will be used to automate the verification process and to ensure that all requirements are met before the degree is verified and added to the blockchain.

5. Transparency and immutability: The blockchain allow to have transparency and immutability of the records, which can be seen by everyone, and it can't be altered in any way once recorded.

6. Auditing and tracking: The system can be used to track and audit the degree verification process, which can help ensure the reliability of the system

7. Multiple verifiers: The system allows multiple verifiers, such as educational institutions and employers, to verify degree information on the blockchain.

3.6.1.1 Technologies to Be Used:

- Front-End: React.JS
- Back-End: Node.JS
- Blockchain: Ethereum

3.7 Alternative Solution

An alternative solution for a degree verification system could be a centralized database system. In this approach, a single entity, such as a government agency or educational institution, would maintain and control a database containing information about degrees and their holders.

3.7.1 Technologies to Be Used:

- Front-End: React.JS
- Back-End: Node.JS
- Database: MongoDB

3.8 External Interface Requirements

3.8.1 User Interfaces

The user interface of Degree verification systems is in blue and white color. It provides various verification techniques to user. It has higher Usability as verifier just to click on any one of option either for document upload, QR-code scan or enter Unique ID. Rest of all is automated and verification is just one click away. On other hand University will be immediately provided with an E-degree and QR-code generated by the system. Once they input all the student's details in text field. User can also use this website from mobile, tablet, computer, and laptop as it is responsive.

3.8.2 Software Interfaces

3.8.2.1 Web browser:

To access the web interface of the system, users will need to have a web browser installed on their device.

3.8.2.2 Wallet software:

Users may need to have a digital wallet software installed on their devices to store their private and public keys and interact with the blockchain network.

3.8.2.3 Meta Mask or similar browser extension:

Meta Mask is a browser extension that allows users to interact with decentralized applications (DApp) on the Ethereum blockchain. If the system is built on Ethereum, users might need to have a similar extension installed in their browser

3.8.2.4 Ledger explorer:

To explore the blockchain ledger, users might need a specific explorer that is built for the blockchain used in the system.

3.8.3 Communications Protocols

3.8.3.1 Blockchain network protocols:

The system would need to use the communication protocols specified by the blockchain platform it is built on, such as the peer-to-peer protocol used by the Bitcoin network or the devp2p protocol used by the Ethereum network.

4. SOFTWARE DESIGN DESCRIPTION

4.1 Introduction

A software design description for a degree verification and generation system using blockchain describes the design of a software system, including the architecture, components, interfaces, and other design elements. The SDD provides a detailed overview of how the software is organized and how it will function, and it serves as a reference for developers and other stakeholders during the development and maintenance of the software.

4.2 Design Overview

The system does not contain ER diagram because of Mongo database. Mongo dB does not use the traditional SQL tabular relational database structure, it uses documents with optional schemas. As we are using Agile Methodology for this purpose, so the system must be flexible enough for applying changes to fulfill the needs of the clients.

4.3 Work Flow Diagram

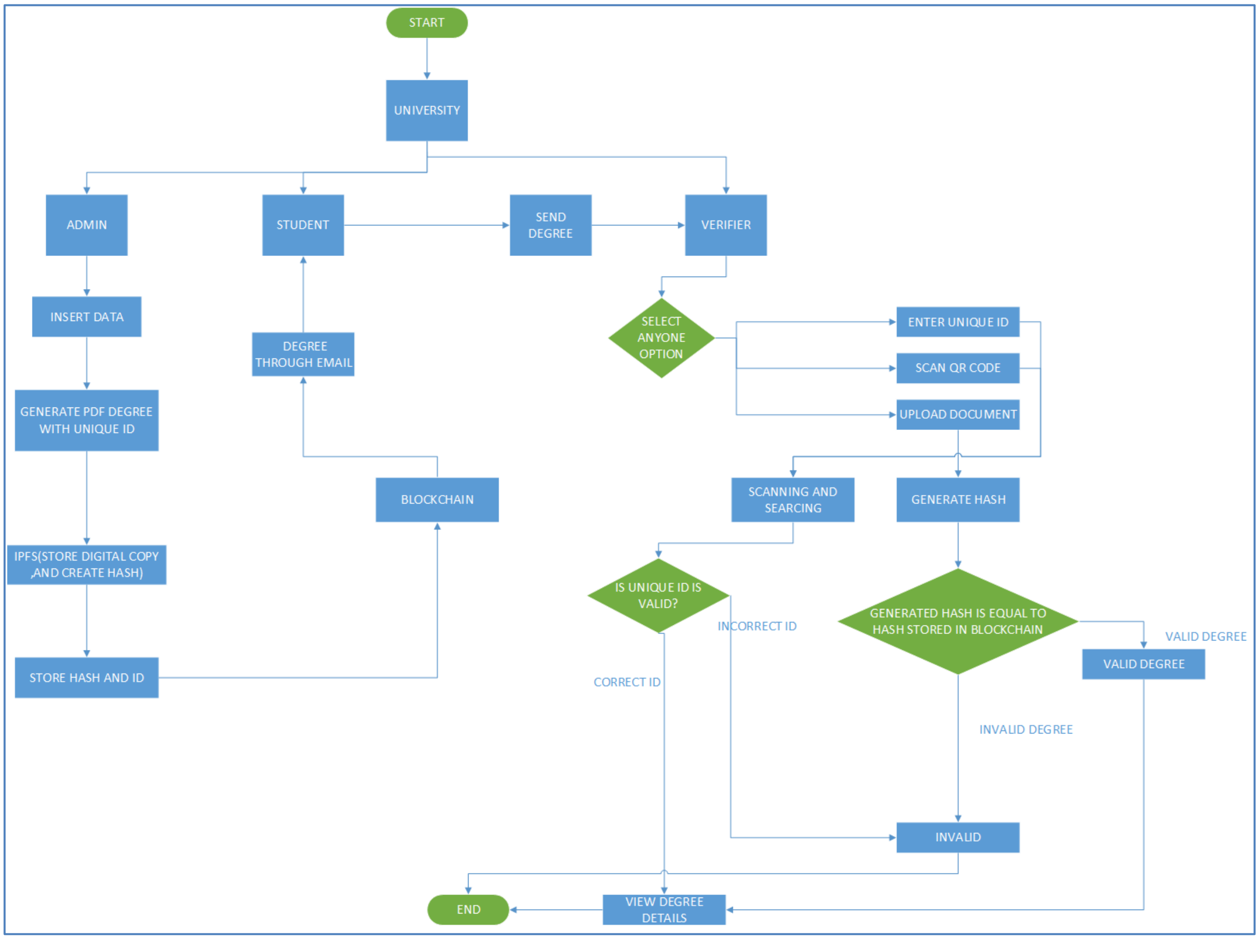


Figure 6: Workflow Diagram

4.4 Business Workflow Diagram

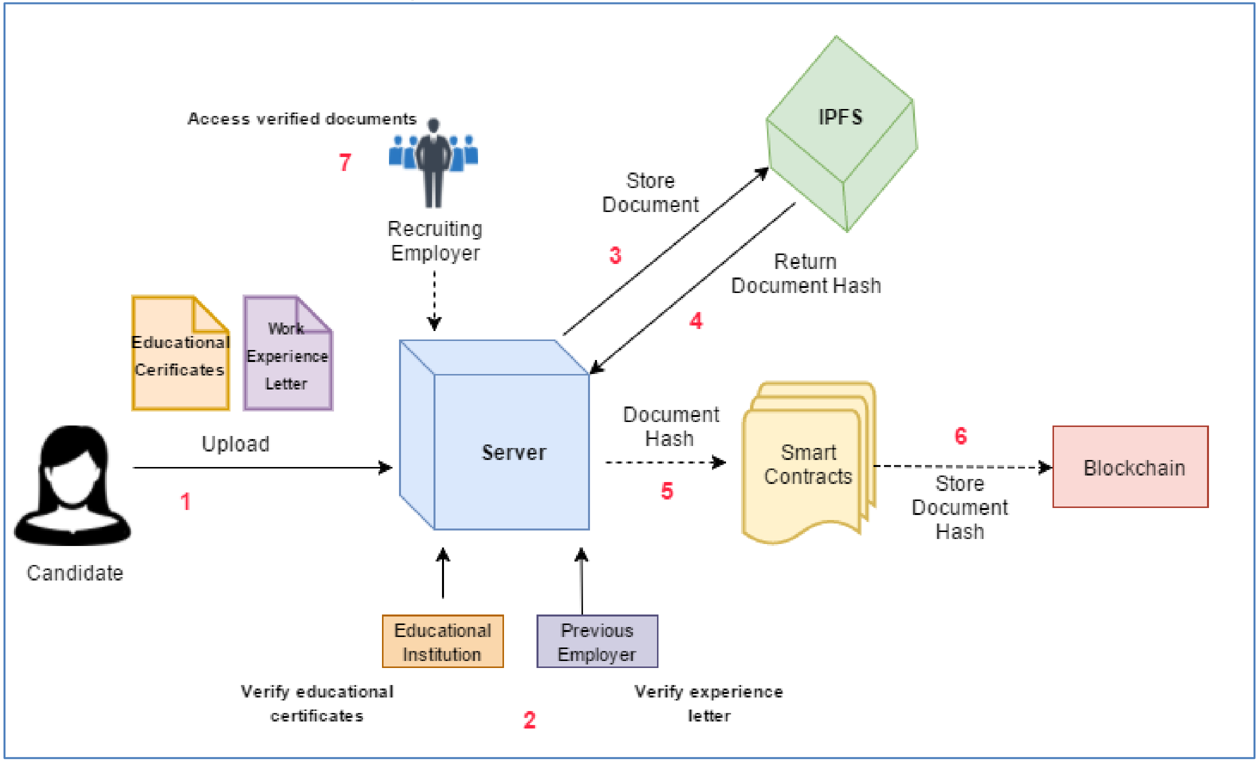


Figure: 6 Business Workflow

4.5 Work Breakdown Structure

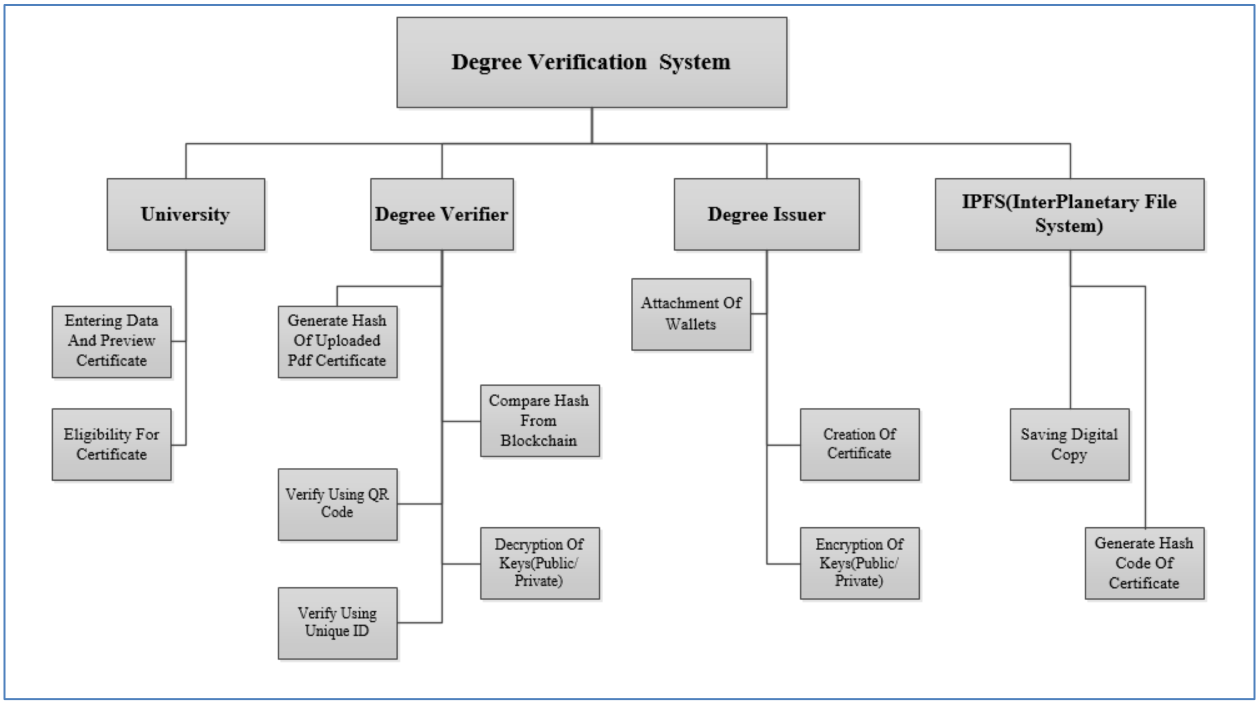

Degree Verification System
University
Degree Verifier
Degree Issuer
IPFS(InterPlanetary File System)
Entering Data And Preview Certificate
Eligibility For Certificate
Generate Hash Of Uploaded Pdf Certificate
Compare Hash From Blockchain
Verify Using QR Code
Decryption Of Keys(Public/ Private)
Verify Using Unique ID
Attachment Of Wallets
Creation Of Certificate
Encryption Of Keys(Public/ Private)
Saving Digital Copy
Generate Hash Code Of Certificate

Figure: 7 WBS

4.6 Use case Diagram

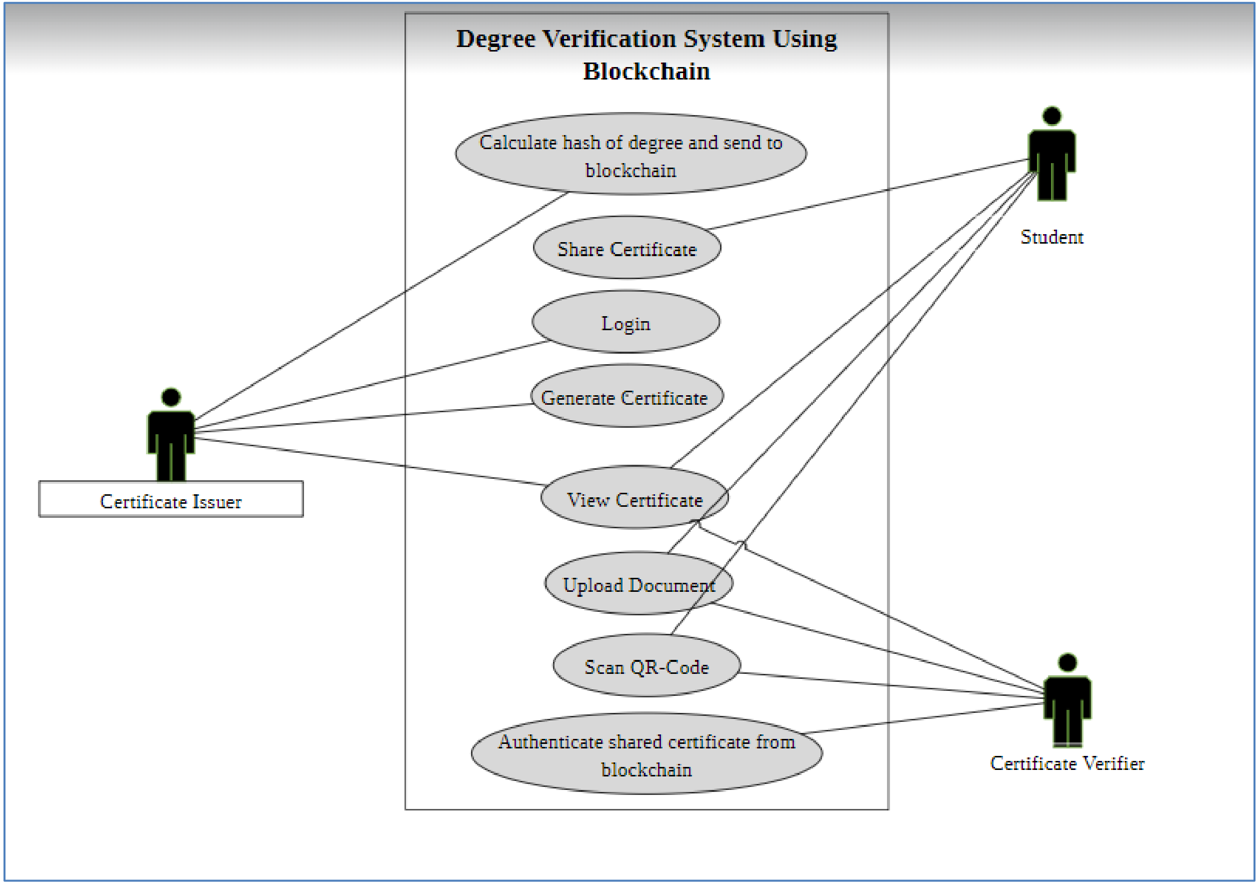

Degree Verification System Using Blockchain
Calculate hash of degree and send to blockchain
Share Certificate
Login
Generate Certificate
View Certificate
Upload Document
Scan QR-Code
Authenticate shared certificate from blockchain
Student
Certificate Issuer
Certificate Verifier

Figure 8: Use Case Diagram

4.7 Sequence Diagram

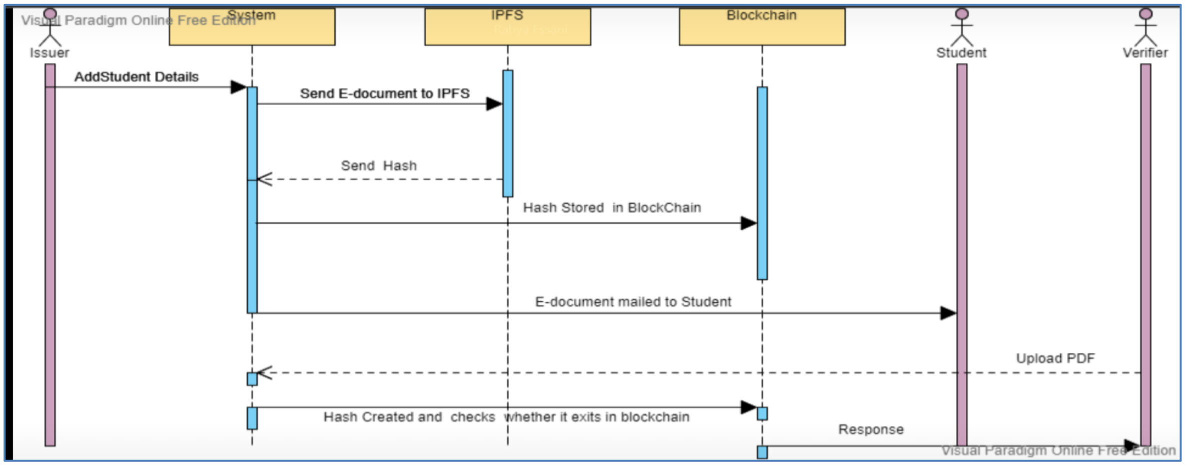

Visual Paradigm Online Free Edition
System
IPFS
Blockchain
Issuer
Student
Verifier
AddStudent Details
Send E-document to IPFS
Send Hash
Hash Stored in BlockChain
E-document mailed to Student
Upload PDF
Hash Created and checks whether it exits in blockchain
Response
Visual Paradigm Online Free Edition

Figure 9: Sequence Diagram

4.8 Requirements Traceability Matrix

Provide a matrix showing where each feature identified in the SRS is supported by the design components.

		Components				
		1	2	3	4	5
Requirements	Authentication	Login				
	Secure and tamper-proof storage of degree information		Certificate Issuer	View Certificate		
	Encryption and digital signature		Certificate Issuer			
	Searching and retrieving			QR-code	Unique ID	Upload Document

If you are developing the software in multiple increments, then a traceability matrix should be produced for each version

4.9 System Architectural Design

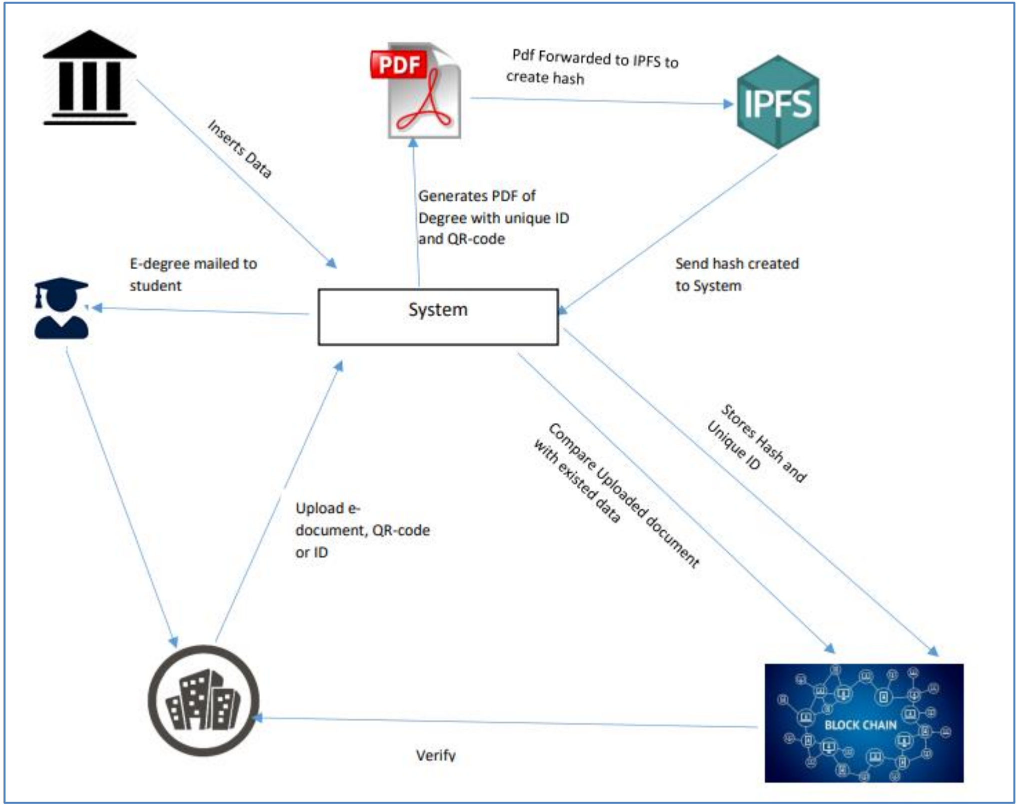

PDF
Pdf Forwarded to IPFS to create hash
IPFS
Inserts Data
Generates PDF of Degree with unique ID and QR-code
E-degree mailed to student
Send hash created to System
System
Stores Hash and Unique ID
Compare Uploaded document with existed data
Upload e-document, QR-code or ID
BLOCK CHAIN
Verify

Figure 10: System Architectural Design

4.10 Chosen System Architecture

Describe the system architectural design, identifying the major component groupings and the interfaces (both internal and external). Make sure to identify any significant technical risks, and identify contingency plans for each.

The University Degree Issuer would use a web application to enter student information, which would be used to create an electronic degree (E-degree). The E-degree would then be sent to the issuer for confirmation, and once confirmed, a QR code would be generated. The E-degree would then be sent to IPFS to generate a hash, which would be sent back to the system and added to the blockchain for storage. The E-degree would then be emailed to the student.

The student could then forward the E-degree to any organization for verification. The verifier would have the option to scan the QR code, enter a unique ID, or upload the E-degree. The QR code or unique ID would be checked against the blockchain to confirm the authenticity of the degree. The verifier could also upload the E-degree and compare the hash of the document to the one stored on the blockchain to confirm the authenticity of the degree.

4.10.1 Blockchain Platform:

The core component of the system would be the decentralized blockchain platform, such as Ethereum or Hyperledger, which would be used to store and verify the degree information.

4.10.2 Smart Contracts

Smart contracts would be used to automate the verification process and ensure that all requirements are met before the degree is verified and added to the blockchain.

➢ **Some technical risks that could be identified in this system are:**

1. **Security risks:** the system must be able to protect personal data stored in the system and prevent unauthorized access
2. **Scalability risks:** the system must be able to handle a large amount of data and a large number of users and verifiers.
3. **Interoperability risks:** the system must be able to exchange data with other systems and applications.

➢ **Contingency plans to mitigate these risks could include:**

1. Regularly reviewing and updating security protocols and access controls.
2. Implementing sharding or other scalability solutions.
3. Developing interfaces and APIs to allow for easy integration with other systems.

4.11 Discussion of Alternative Designs

Centralized database: A centralized database, would store information about degrees and their holders. The centralized database would be connected to a decentralized blockchain network, which would be used to store a hash of the degree information in a secure and transparent way. A mechanism would be implemented to securely connect the centralized database and the blockchain network, allowing for data to be transferred and hashed on the blockchain.

We are not using it because the level of transparency would be limited as some data is kept in centralized database and not all parties can access it.

4.12 System Interface Description

Describe the system interfaces in detail: O/S interface, files, networking, libraries, graphics libraries etc. (Describe the user interface in section 4.)
Operating System Interface: The system would need to interface with the underlying operating system to access system resources, such as memory and storage, and perform system-level operations.

File Interface: The system would need to interface with file systems to read and write data to storage, such as storing degree information in a standardized format, such as JSON or XML.

Network Interface: The system would need to interface with networking protocols and communication interfaces to connect to the blockchain network, interact with other systems and applications, and transfer data over the network.

Blockchain libraries: The system would use libraries that provide a higher-level interface to interact with the blockchain platform, such as Geth or Parity for Ethereum.

Cryptography libraries: The system would use libraries to implement encryption and digital signature algorithms to secure the data stored on the blockchain.

API interface: The system would offer APIs to interact with the system, which would allow other systems and applications to interact with the blockchain and verify the degrees stored on it.

4.13 User Interface Design

4.13.1 Description of the User Interface

4.13.1.1 Admin Module

The first step in this process is that an administrator logs into the system and enters information about a student. This information is then sent to the system.

4.13.1.2 System Module

Next, the system generates a certificate with a unique identification number and a QR code and sends the certificate to both the student and the IPFS. On IPFS, the system stores the student's degree and generates a unique hash for the degree, which is then further send to the blockchain.

4.13.1.3 Verifier Module

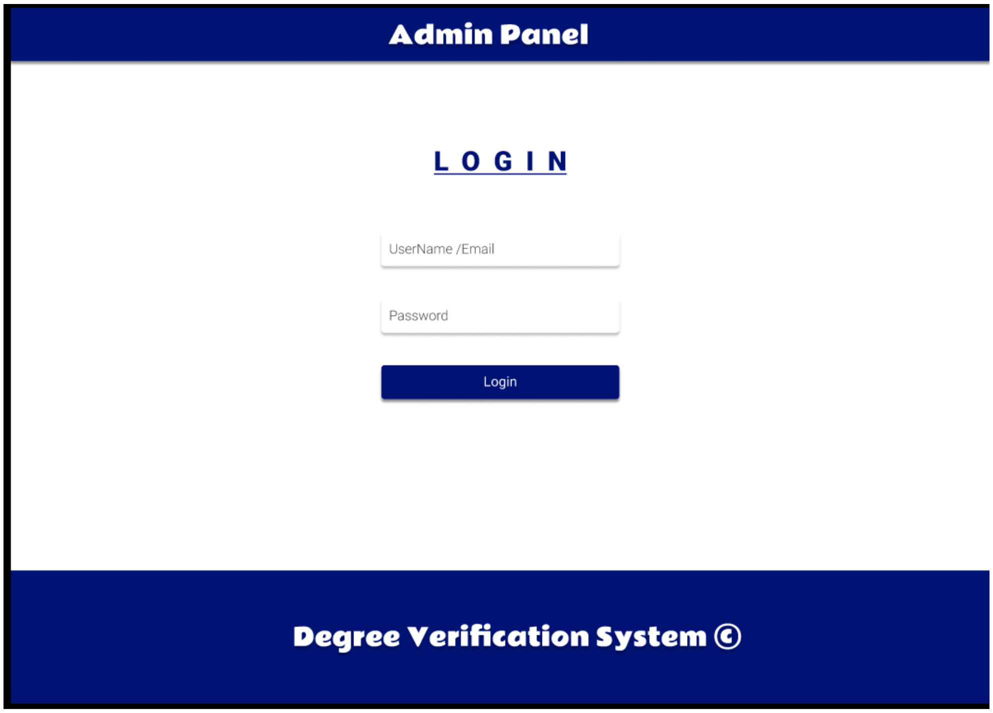
Finally, when a student wants to share their degree with a third party, such as a company or interviewer, the third party has several options for verifying the degree's authenticity. These include:

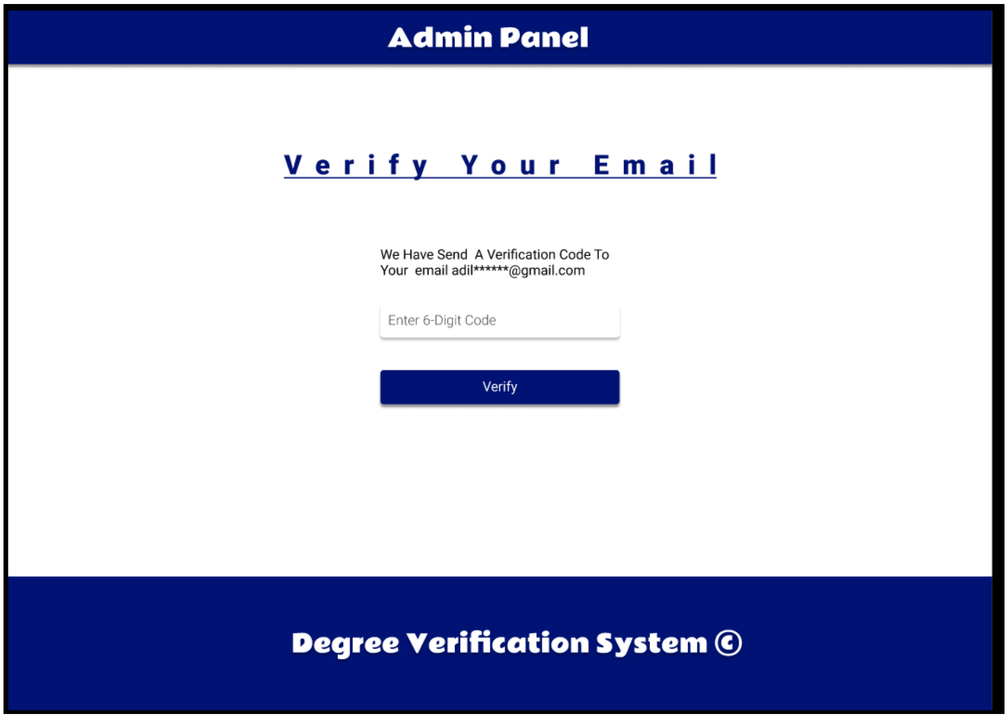
- Uploading the original document,
- Scanning the QR code, or
- Entering the unique identification number

The system then compares the information provided by the third party to the information recorded on the blockchain and notifies the third party if the degree is valid or not.

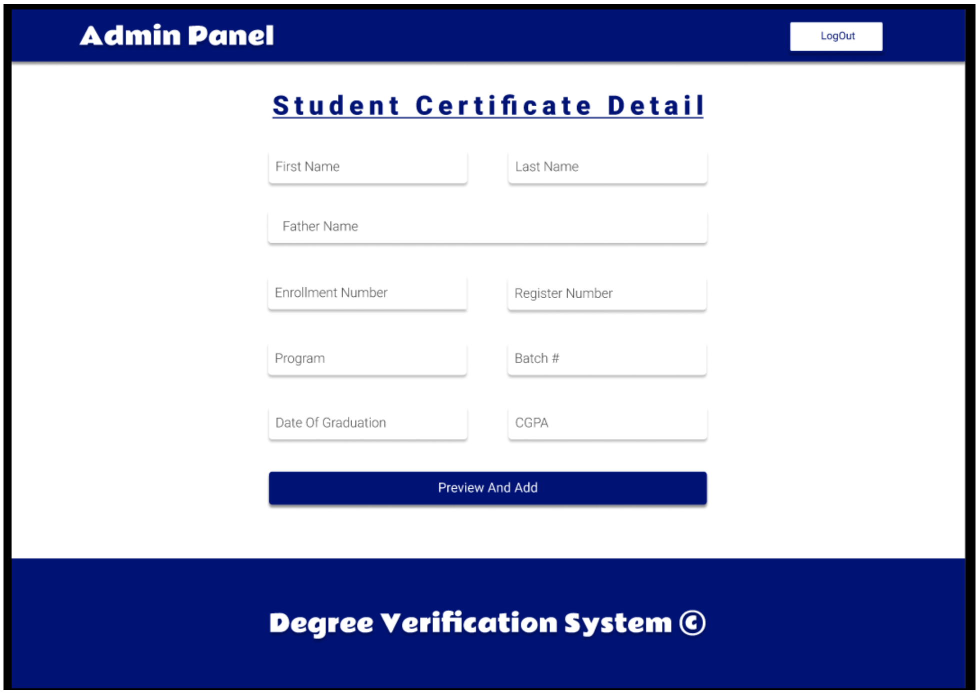
4.13.2 Screen Images

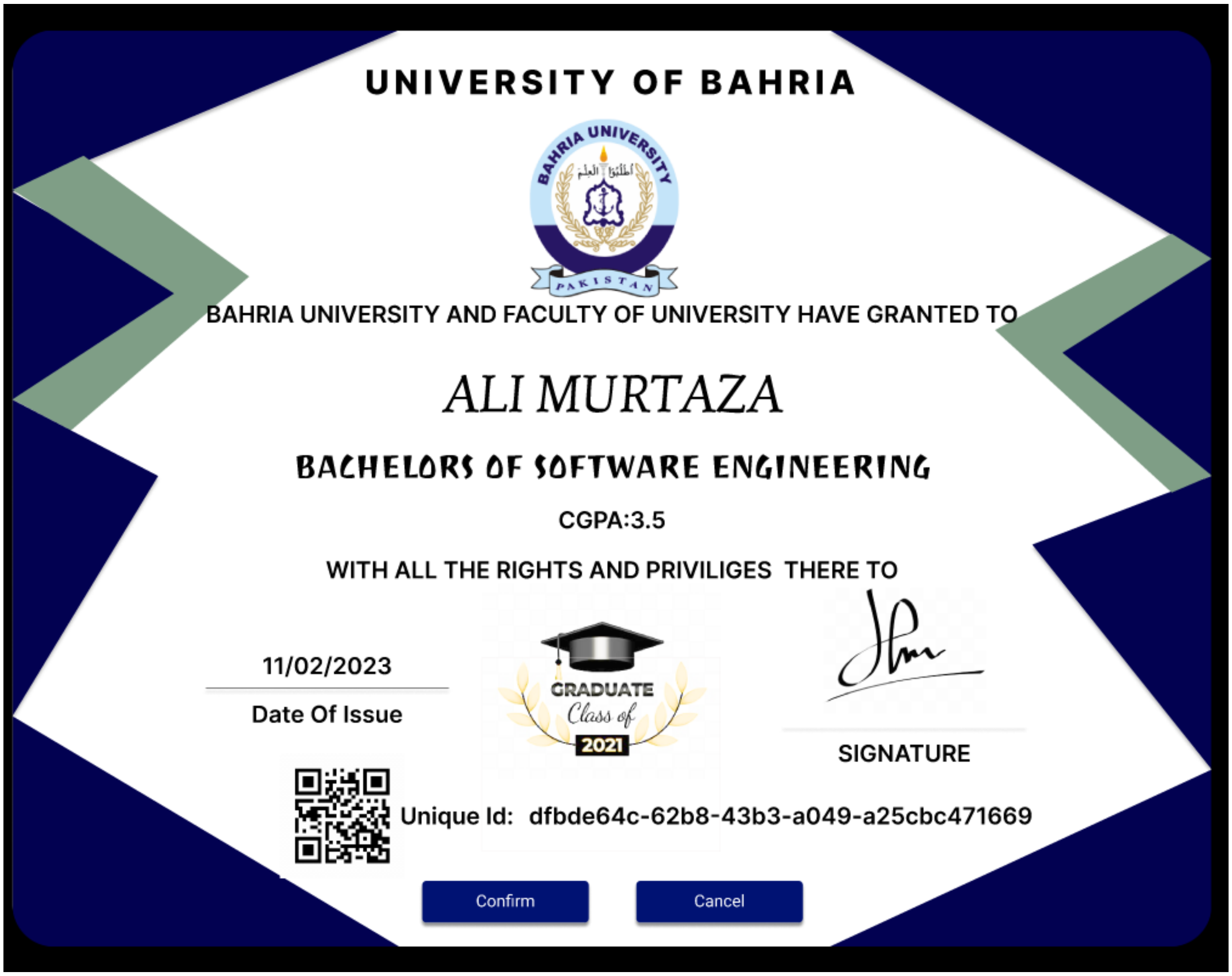
4.13.2.1 Admin Module





We Have Send A Verification Code To Your email adil******@gmail.com





UNIVERSITY OF BAHRIA

BAHRIA UNIVERSITY AND FACULTY OF UNIVERSITY HAVE GRANTED TO

ALI MURTAZA

BACHELORS OF SOFTWARE ENGINEERING

CGPA:3.5

WITH ALL THE RIGHTS AND PRIVILIGES THERE TO

11/02/2023

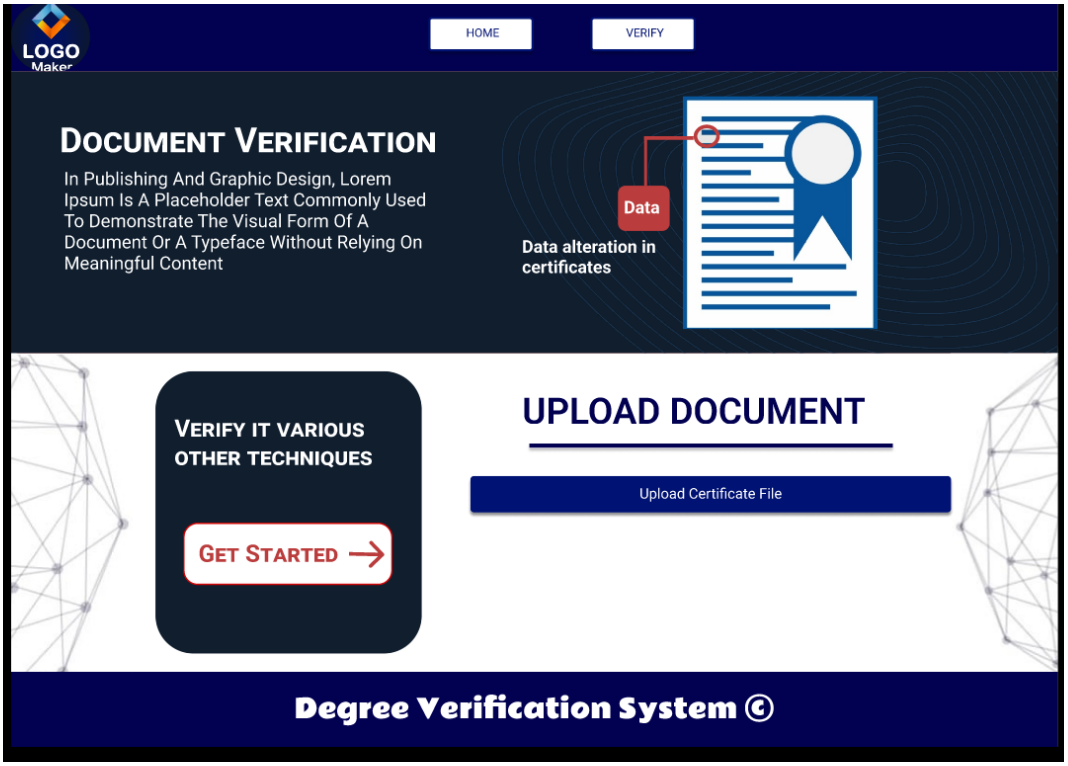
Date Of Issue

GRADUATE Class of 2021

SIGNATURE

Unique Id: dfbde64c-62b8-43b3-a049-a25cbc471669

4.13.2.2 Public Accesible Module



DOCUMENT VERIFICATION

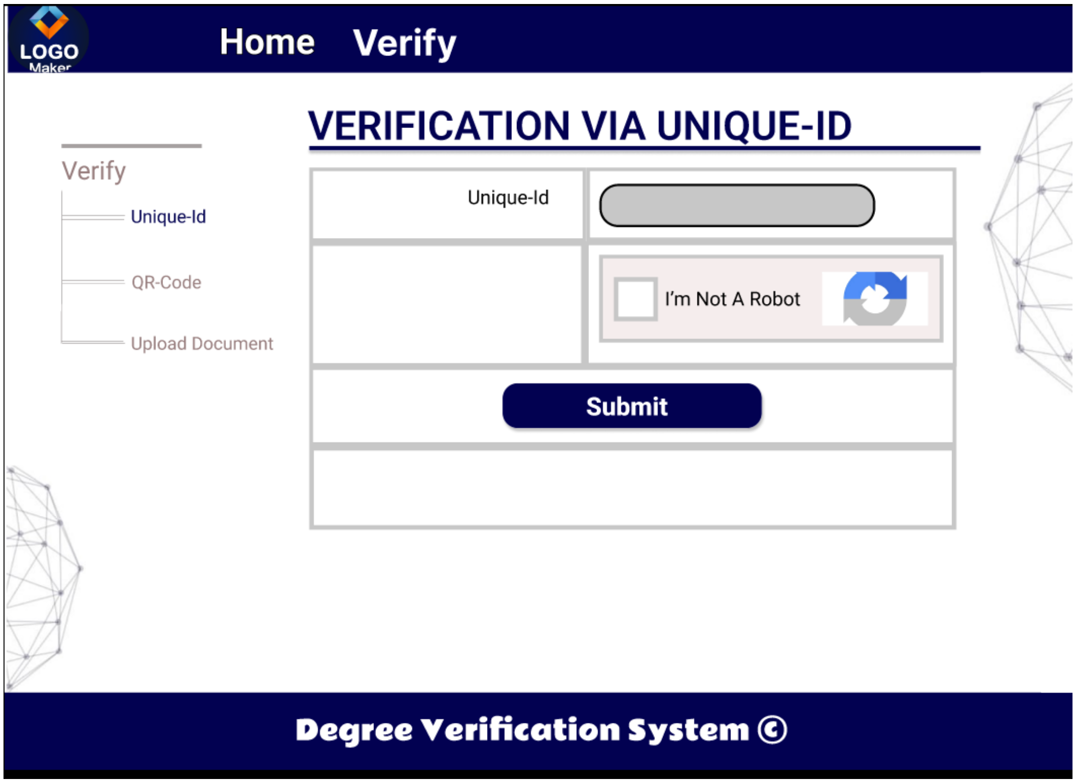
In Publishing And Graphic Design, Lorem Ipsum Is A Placeholder Text Commonly Used To Demonstrate The Visual Form Of A Document Or A Typeface Without Relying On Meaningful Content

Data alteration in certificates

VERIFY IT VARIOUS OTHER TECHNIQUES

UPLOAD DOCUMENT

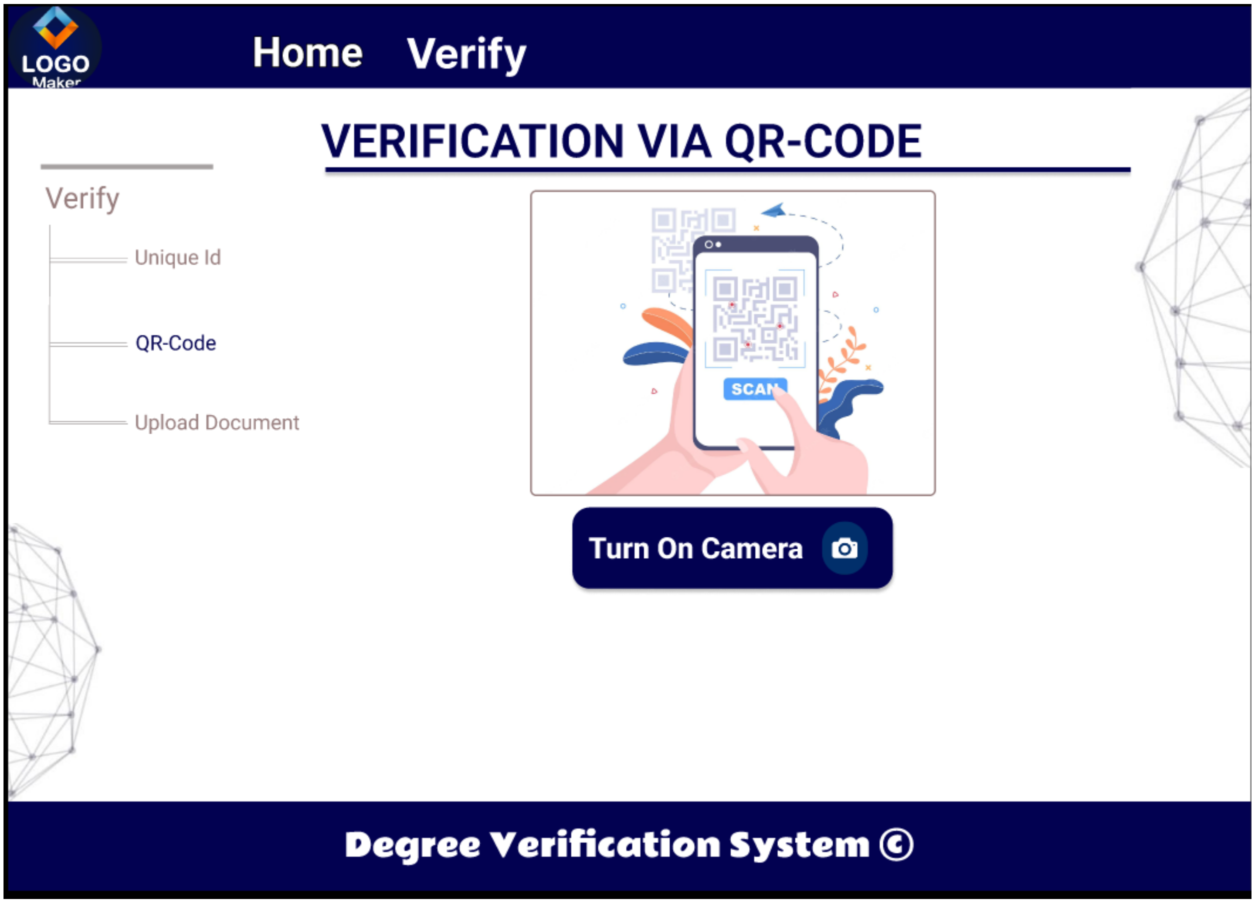
Degree Verification System ©



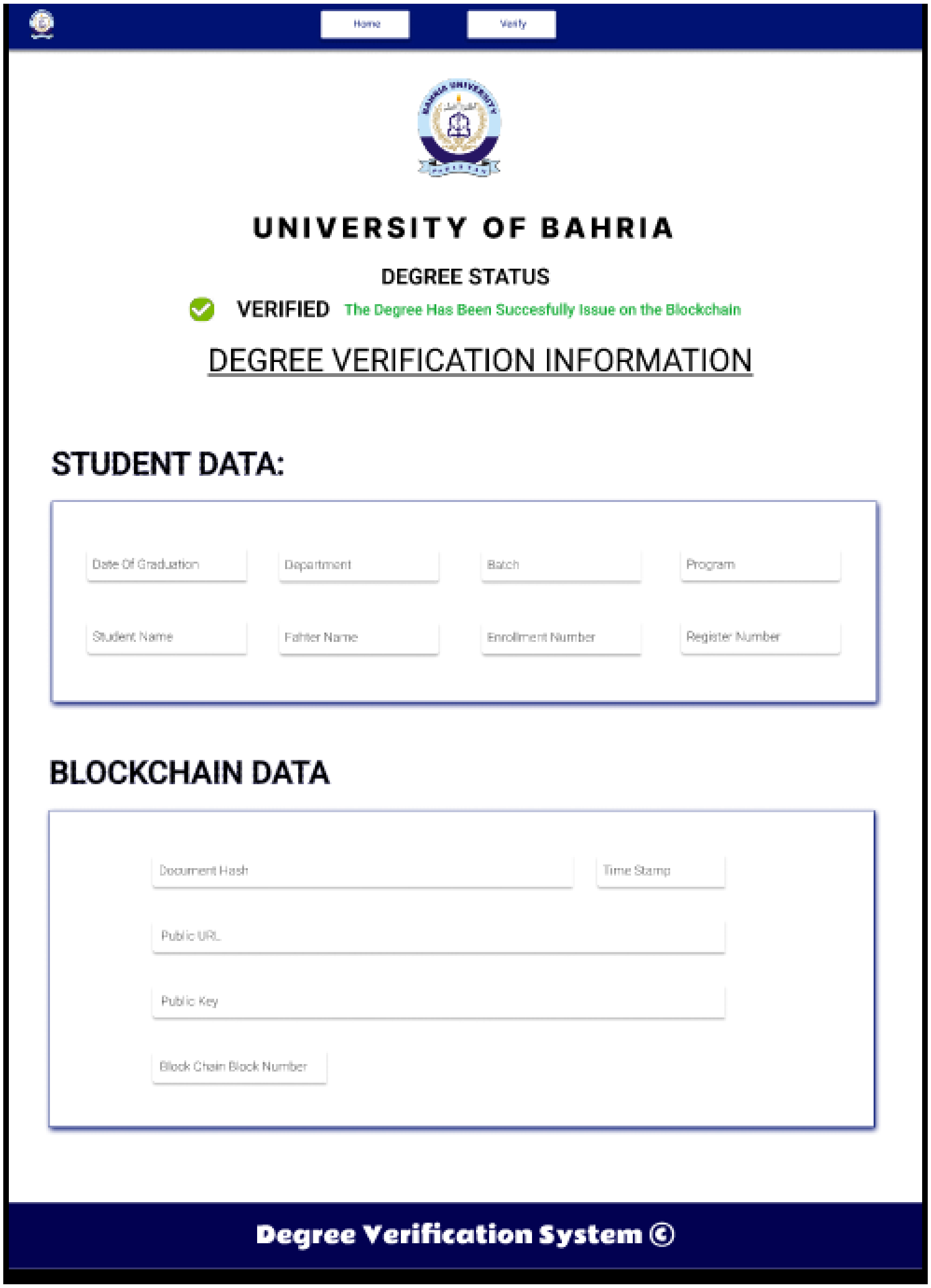
VERIFICATION VIA UNIQUE-ID

Unique-Id	
	□ I'm Not A Robot

Degree Verification System ©



LOGO
Maker
Home
Verify
VERIFICATION VIA QR-CODE
Verify
Unique Id
QR-Code
Upload Document
SCAN
Turn On Camera
Degree Verification System ©



UNIVERSITY OF BAHRIA

DEGREE STATUS

✓ VERIFIED The Degree Has Been Succesfully Issue on the Blockchain

DEGREE VERIFICATION INFORMATION

STUDENT DATA:

BLOCKCHAIN DATA

Degree Verification System ©

4.13.3 Objects and Actions

The system includes a text field for the administrator to enter data about the student. The user interface also includes various buttons, labels, icons, images, and anchor tags to help the administrator, verifier, student navigate and use the system. The design of the system is responsive and well-organized, making it easy for the administrator, student, and verifier to use. The buttons in the UI can be used to make selections, icons can be used to represent different actions, and labels can be used to provide context. Images and anchor tags can be used to enhance the user experience or provide extra functionality. Overall, the system is designed to provide a clear and intuitive experience for all users

5. IMPLEMENTATION

This chapter must contain any of the following that you have use to address/solve the problem:

- pseudo codes
- algorithms
- actual code

This section must not include auto generated code. List only your written code. Give code details (not a complete listing, but description of key parts). Discuss the most important/interesting aspects. It probably won't be possible to discuss everything- give a rationale for what you do discuss. Code shall not be more than 3-5 pages.

6. SOFTWARE TEST DOCUMENT

This system test document specifies the tests for the entire software system, defines the test schedule, and records the test results.

This document does not cover unit testing (the testing of individual sub-systems or components of the system).

The system may consist of multiple items that are to be tested separately.

The system may be tested in one or more increments of functionality; the system test document should cover each version of the system separately. When different versions of the system are tested, make sure to clearly identify the version of the software and the relevant test and test result. Also, extend the unique identifier scheme to include the version of the software system under test (SUT) – for example, use TC-vvvv-nnnn to identify test case nnnn for software system version vvvv.

6.1 System Overview

Briefly detail the software system and items to be tested. Identify the version(s) of the software to be tested.

6.2 Test Approach

Describe the overall approach to testing. For each major group of features or feature combinations, specify the approach that will ensure that these feature groups are adequately tested. Specify the major activities, techniques, and tools that are used to test the designated groups of features. The approach should be described in sufficient detail to permit identification of the major testing tasks and estimation of the time required to do each one. Identify significant constraints on testing, such as deadlines.

6.3 Test Plan

Describe the scope, approach, resources, and schedule of the testing activities. Identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task in the case of a group project.

6.4 Features to be Tested

Identify all software features and combinations of software features to be tested. Identify the test case(s) associated with each feature and each combination of features. Identify the version of the software to be tested.

When multiple versions of the software are tested in a planned, incremental manner, then use section numbers 2.1.n to identify the features to be tested for each version.

6.5 Features not to be Tested

Identify all features and significant combinations of features that will not be tested and the reasons for not doing so.

6.6 Testing Tools and Environment

Specify test staffing needs. For an individual project, specify the time to be spent on testing. For a group project, specify the number of testers and the time needed.

Specify the requirements of the test environment: space, equipment, hardware, software, special test tools. Identify the source for all needs that are not currently available.

6.7 Test Cases

A test case specification refines the test approach and identifies the features to be covered by the case. It also identifies the procedures required to accomplish the testing and specifies the feature pass/fail criteria. It documents the actual values used for input along with the anticipated outputs.

If an automated test tool is to be used:

1) Document each test case here as a specification for the test tool;
2) Document the procedure that must be followed to use the test tool.

6.8 Case-n (use a unique ID of the form TC-nnnn for this heading)

6.8.1 Purpose

Identify the version of the software and the test items, and describe the features and combinations of features that are the object of this test case. For each feature, or feature combination, a reference to its associated requirements in the software requirement specification (SRS) should be included.

6.8.2 Inputs

Specify each input required to execute the test case. Some of the inputs will be specified by value (with tolerances where appropriate), while others, such as files or URLs, will be specified by name. Specify all required relationships between inputs (e.g., ordering of the inputs).

6.8.3 Expected Outputs & Pass/Fail criteria

Specify all of the expected outputs and features (e.g., response time) required of the test items. Provide the exact value (with tolerances where appropriate) for each required output or feature. Specify the criteria to be used to determine whether each test item has passed or failed testing. If an automated test tool is used, identify how the results of that tool are to be analyzed.

6.8.4 Test Procedure

Detail the test procedure(s) needed to execute this test case. Describe any special constraints, such as: special set up, operator intervention, output determination procedures, and special wrap up.

7. APPENDIX A: TEST LOGS

A test log is used by the test team to record what occurred during test execution.

7.1 Log

For test-1 (use a unique ID of the form TL-nnnn for this heading)

7.1.1 Test Results

For each execution, record the date/time and observed results (e.g., error messages generated). Also record the location of any output (e.g., window on the screen). Record the successful or unsuccessful execution of the test.

7.1.2 Incident Report

(Add a unique ID of the form TIR-nnnn to this heading)

If the test failed, or passed with some unusual event, fill in this incident report with the details. Summarize the incident, identifying the test items involved, and the anomaly in the results. Indicate what impact this incident will have on the project.

8. CONCLUSIONS AND FURTHER WORK

In this report we have presented explanations for the student software engineering documentation templates introduced in TR05. These explanations are based on the IEEE standards. Their purpose is to guide the students through the process of documenting their software development, and raise their awareness of certain key issues in software engineering.

We recognize that each software project is unique and so we do not advocate using a specific software engineering process based on these templates. In the future we hope to find time to document approaches that students take in producing software and identify the most common processes and pitfalls.

9. REFERENCES