

SIGNIFIED

Final Project Report



Advisor

Dr Hasan Ali Khattak

Co-Advisor

Dr Ahsan Saadat

May 16, 2022

Group Members

Anam Tariq (282737)

Saira Sarwar (284606)

National University of Sciences and Technology, Islamabad

Table of Contents

1. Background	5
1.1 Abstract	5
1.2 Introduction	5
1.3 Literature Review	6
1.4 Problem Statement	9
1.5 Proposed Solution	10
1.6 Project Scope	12
2. Software Requirements Specification	12
2.1 Introduction	12
2.2 Purpose	13
2.3 Document Conventions	13
2.4 Intended Audience and Reading Suggestions	13
2.5 Overall Description	14
2.5.1 Product Perspective	14
2.5.2 Product Functions	14
2.5.3 User Classes and Characteristics	15
2.5.4 Operating Environment	15
2.6 External Interface Requirements	17
2.6.1 User Interfaces	17
2.6.2 Hardware Interfaces	17
2.6.3 Software Interfaces	17
2.6.4 Communications Interfaces	18
2.7 System Features	18
2.7.1 Registration	18
2.7.2 Login	19
2.7.3 User Profile	19
2.7.4 Upload Document	20
2.7.5 Verify Document	20
2.7.6 View Document	21
2.8 Other Nonfunctional Requirements	21
2.8.1 Performance Requirements	21
2.8.2 Safety Requirements	22
2.8.3 Security Requirements	22
2.8.4 Software Quality Attributes	23
2.9 Other Requirements	23
2.10 Use Case Diagram	24
3. Software Document Specification:	26
3.1 Introduction	26
3.2 Purpose	27
3.3 Product Scope	27
3.3 Definitions, acronyms and abbreviations	28
3.4 Overview of the Document	29
3.5 Design Considerations	29
3.5.1 Overview	29

3.5.2	Design Methodology.....	29
3.5.3	Design Models	30
3.6	System Architecture.....	33
3.6.1	Overview	33
3.6.2	System Architecture Description	33
3.6.3	Process Flow.....	36
3.7	Database Schema	38
3.7.1	Tables, Fields and Relationships.....	38
3.8	Data Representation Diagram.....	39
4.	Software Testing.....	41
4.1	Introduction.....	41
4.2	Scope.....	41
4.2.1	Functions to be tested.....	41
4.2.2	Functions not to be tested	41
4.3	Quality Objectives	42
4.4	Test Approach.....	42
4.5	Test Strategy	42
4.6	Test Cases	43
4.6.1	Unit Testing	43
4.6.2	System Testing.....	62
4.6.3	Performance Testing.....	66
5.	Implementation and Deployment	67
5.1	Tools and Software Used	67
5.1.1	Truffle	67
5.1.2	Ganache.....	68
5.1.3	MetaMask	68
5.1.4	React Js	69
5.1.5	Node Js.....	69
5.1.6	PostgreSQL.....	70
5.1.7	AWS S3 Bucket	70
5.1.8	Deployment on S3 Bucket	71
6.	User Interface Design	71
6.1	Home Page	71
6.2	Sign Up Page.....	72
6.3	Login In Page.....	73
6.4	Student Dashboard Page	74
6.5	New Application Page	75
6.6	View Application Page	80
6.7	Verifier Dashboard Page.....	82
6.8	Verified Applications Page.....	82
6.9	Verifier View Application	83
6.10	View Document through Hash Value.....	85
6.11	View Document though Deep Link or QR Code.....	85
	Appendix A: Glossary.....	86
	References	87

Table of Figures

Figure 1: Use Case Diagram of Student	24
Figure 2: Use Case Diagram of Verifier	25
Figure 3: Use Case Diagram of Viewer	25
Figure 4: Use Case Diagram of System.....	26
Figure 5: Class Diagram	31
Figure 6: State Transition Diagram.....	32
Figure 7: System Architecture	35
Figure 8: Activity Diagram	37
Figure 9: Database Tables	39
Figure 10: Entity Relationship Diagram	40
Figure 11: Unit Testing	61
Figure 12: Performance Testing.....	67
Figure 13: Documents Uploaded on AWS S3 Bucket	71
Figure 14: Home page.....	72
Figure 15: Sign up page	73
Figure 16: Login page.....	74
Figure 17: Student Dashboard page.....	75
Figure 18: Personal Details.....	76
Figure 19: Document Details	77
Figure 20: Document Upload.....	78
Figure 21: Preview Details.....	79
Figure 22: Submission	80
Figure 23: View Application Page	81
Figure 24: Verifier Dashboard Page.....	82
Figure 25: Verified Applications Page	83
Figure 26: Verifier View Application.....	84
Figure 27: View Document through Hash Value	85
Figure 28: View Document through Deep Link or QRCode	86

SIGNIFIED

Decentralized Ledger for Document Verification

1. Background

1.1 Abstract

The verification of documents is a complex field that includes numerous challenging processes. It involves verifying various kinds of documents, such as government documents, banking documents, transaction documents, and educational certificates, each of which may require distinct verification methods. Out of these documents, educational certificates issued by universities are the most important for students. However, the issuing process is not always transparent or verifiable, making it easy to create fake certificates. Consequently, these documents must undergo verification by an authorized party, which can be a lengthy process. Recently, Blockchain technology has emerged as a promising means of authenticating the document verification process and combating document fraud and misuse.

1.2 Introduction

In Pakistan, the education system typically involves students enrolling in kindergarten, then changing schools for primary, secondary, and high school studies. After completing high school, students must get admitted to a university, and some students may change colleges for post-graduate studies.

At each stage of this cycle, students receive various certificates. However, when students apply for higher education abroad, their certificates must be verified by the Higher Education Commission of Pakistan, which can be a time-consuming and risky process.

Pakistan has a large population and millions of students graduate each year, it is a challenging task to keep track of and verify each certificate which leads to fraudulent certificates and degrees. To tackle this issue, Blockchain technology has emerged as a potential solution. So first let us understand what is meant by Blockchain.

- **Blockchain:**

The concept of blockchain was proposed by Satoshi Nakamoto in 2008. Blockchain is an online ledger that provides decentralized and transparent data sharing. Blockchain technology is a solution for the problem of centralization. It is a system of keeping records by everyone without any need of a central authority - a decentralized way of maintaining a ledger that is practically impossible to falsify.

So, the next important thing is how the blockchain technology actually works. Well, there are four main elements of blockchain. The first element required to support a blockchain is a peer-to-peer network. The second element is cryptography. The third element is a consensus algorithm. Finally, the last element is punishment and reward. Our proposed system uses Ethereum Blockchain technology and solves the issue students face in document verification.

1.3 Literature Review

The project mainly focuses on building a blockchain based solution for online documents verification. For this, we have referred to some previously published papers and works of the various individuals in this field.

- **Published Papers:**

- 1. BlockIPFS - Blockchain-Enabled Interplanetary File System for Forensic and Trusted Data Traceability [1]:**

- i. The research paper above basically discusses about IPFS and its integration with Blockchain. It actually compared the traditional IPFS with Blockchain and the results that were produced won in most categories, such as reading transactions, uploading transactions, and downloading the transactions [1].

2. Tamper Proof Birth Certificate Using Blockchain Technology [2]:

This paper focuses on the tamper proof birth certificates and running the AES algorithm and storing the data on IPFS to secure the records as well as to access and share records with user permission form anywhere in the network [2].

3. Framework Authentication e-document using Blockchain Technology on the Government system [3]:

The focus of this research paper is to utilize Blockchain technology to maximize the government documents in a secure manner. It basically proposes a Blockchain based framework to ensure the speed of the system by utilizing the DAO (Decentralized Autonomous Organization) and Smart Contracts. As a result, the system can significantly maintain transparency and increase trust in public services [3].

● Existing works related to Proposed Solution

So, in this section of literature review we have mentioned some of the existing works related to the proposed system.

1. First e-Verified Certification Platform in Hong Kong launched by HKUST [4]:

The Hong Kong University of Science and Technology (HKUST) has launched the HKUST “Blockcerts” platform which is a trust-free and user-friendly verification system for documents and this also promotes paperless and sustainable [4].

HKUST is the first university in Hong Kong which is replacing the traditional verification of papers with the Blockchain based system where students make requests for the verification, pay the verification fee and the prints the copy of the degree. The system offers environment friendly solution and it also saves a lot of time of students as well. [4].

2. MyDiplome [5]:

It is the new CIMEA certification service using blockchain technology: the certificates are saved in blockchain: allowing the holders of the qualifications to share their academic titles with institutions and employers in a simple, secure and certified way.

So, let's start with what CIMEA is and what its procedure is. So basically CIMEA, first started its performance in 1984 and advises the procedures of qualifications recognition. Its primary objective is to propose mobility in different aspects such as facilitating the elements of Italian system [5].

It has introduced a wallet for each account holder where they upload their documents. They have introduced a decentralized, transparent and unchangeable system. It is actually a global ecosystem which provides security as well as guaranteeing the authenticity [5].

In Diplome the qualifications and certificates are saved by the institutions that issue them directly on blockchain, guaranteeing the unchangeability and transferability of the data, security and simplicity in sharing, protection and respect of personal data.

3. University of Sharjah - to utilize BSV blockchain to verify academic certificates [6]:

The BSV conferences hosted by Coin Geek are incredible gatherings that cover a wide range of subjects relevant to blockchain, Bitcoin, and practically every industry worldwide. The Head of Technology Transfer Office of the University of Sharjah, Dr. Mohamed Al Hemaury, gave a presentation titled "Academic Accreditation & Certification on the BSV Blockchain" on Day 1 of the first-ever BSV Global Blockchain Convention in Dubai [6].

Dr. Al Hemaury addressed the issue of unrestricted use of bogus diplomas and certificates across the globe, which is frequently overlooked, whether intentionally or unintentionally. The University of Sharjah is working with the BSV Blockchain Association on a project to address this issue, with the goal of eradicating this unethical practice by transferring the process of issuing and validating all academic diplomas to the BSV enterprise blockchain [6].

4. MIT Media Lab - Blockcerts Blockchain Credential Certification [7]:

MIT Media Lab released Blockchain Certificates, an open standard for digital academic certificates on the Bitcoin blockchain. The platform also allows a decentralized credentialing system. The application can be used for academic, as well as professional, and workforce credentialing. The distributed ledger is fully transparent, and as such these data can be publicly shared for verification. It builds an ecosystem for creating, sharing, and verifying blockchain-based educational certificates. Digital certificates are registered on the Bitcoin blockchain, cryptographically signed, and tamper proof [7].

They are actively looking for collaborators to build out a few example implementations outside the Media Lab. Areas in which digital certificates provide exciting opportunities include Corporate/ enterprise training and Workforce development.

1.4 Problem Statement

For students, educational certificates are the most important documents issued by their universities. However, fake certificates can be easily created and it is always hard to detect and can be treated as the original. So, students need to verify their documents from HEC. Document verification is essentially needed to prove the legitimacy of the documents for official or legal purposes. The current process of document verification is very lengthy and comprises of the following steps:

- The applicants need to apply online for degree attestation.
- The initial examination by HEC can take 8 to 10 days depending on the workload.
- After successful online examination by HEC, applicants are required to schedule their appointment and wait for their turn.
- They are required to visit the HEC office along with their documents for verification.
- This process requires time and other resources.

1.5 Proposed Solution

Our proposed solution is an automated verification system which can reduce the time required for the document verification process. We aim to develop a blockchain based solution for online documents verification. Using the Ethereum Blockchain technology, we can achieve a decentralized system to verify documents without the need of a certain level of human intervention. This will drastically improve the document verification process's efficiency along with the benefits of security, reliability and transparency and make document verification process easier, safer and more convenient for users.

The participants which will interact with our proposed solution are as follows:

1. Students:

Students will be able to upload digital documents and fill in the details required for successfully uploading documents.

2. Verifying Authority:

Verifying authority will be the user who will verify the documents uploaded by students. They will provide originality, authenticity, and integrity to the documents. After verification, documents will have a hash value which in the future can be used to access the document.

3. Viewers:

Viewer will be the user who can see the verification process between the student and verifying authority using the hash value of the document.

Advantages of Proposed Solution:

Our proposed solution consists of following advantages:

1. Reliable System:

Blockchain technology provides a reliable system because the data stored on a Blockchain cannot be realistically altered. Any attempts to modify the data can be easily detected, ensuring the integrity and security of the information.

2. Speedy Process:

Another benefit of using Blockchain technology for document verification is that the process is much faster than traditional methods. With automation, document verification can be completed in a matter of seconds, eliminating the time-consuming and often manual processes associated with traditional verification methods.

3. Safe Document Handling:

Storing certificates digitally on the Blockchain also eliminates the risk of physical damage or loss. As the certificates are stored digitally and can be accessed from anywhere with an internet connection, students no longer have to worry about losing or damaging their certificates during the verification process.

4. Economical Process:

Implementing a Blockchain-based document verification system can also lead to cost savings as the process will be fully automated. This reduces the need for manual labor and administrative costs associated with traditional document verification methods, resulting in a more cost-effective and efficient system.

The proposed Blockchain-based document verification system not only addresses the loopholes in the current system but also provides a reliable and practical solution to document verification. By leveraging the security and automation features of Blockchain technology, the system can provide a more secure, efficient, and cost-effective solution for document verification, ultimately benefiting students, educational institutions, and other stakeholders involved in the process.

1.6 Project Scope

SIGNIFIED is an innovative and automated verification system that aims to reduce the time required for document verification. Our solution is based on the Ethereum Blockchain technology, which allows us to develop a decentralized system for online document verification, without the need for human intervention. By utilizing the security, reliability, and transparency features of Blockchain technology, SIGNIFIED will drastically improve the efficiency of the document verification process. Our solution will provide a more convenient, safer, and easier way for users to verify their documents, and ultimately offer a more seamless experience.

The process of document verification can be complicated, and it involves several challenging procedures to ensure authenticity. Among the documents that require authentication, educational certificates issued by universities are of great importance. However, the issuing process of these certificates lacks transparency, making it easier to create fake certificates. Additionally, when students apply to study abroad, their certificates need to be verified by HEC, which can be a lengthy process. To address these challenges, Blockchain technology has emerged as a promising means of authenticating the document verification process and combating document fraud and misuse.

2. Software Requirements Specification

2.1 Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the project **SIGNIFIED-Decentralized Ledger for Document Verification** by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by

stakeholders and their needs while defining high-level product features. The detailed requirements of **SIGNIFIED** are also provided in this document.

2.2 Purpose

The purpose of this document is to present a detailed description of SIGNIFIED-Decentralized Ledger for Document Verification. It will describe the functions and characteristics of the system, its interfaces, what the system will perform, the limitations that must be met for it to function, and how the system will respond to outside stimuli. This document will be submitted to the higher authorities for approval and is meant for both the stakeholders and the system developers.

2.3 Document Conventions

The document focuses on the high priority requirements which will be implemented for the final deliverable. The document is organized in accordance with the Software Requirements Specification template of IEEE. Bold-faced text has been used to emphasize section and subsection headings. These headings are written in Times New Roman with font size 14. However, the rest of the document is written in Times New Roman with font size 12.

2.4 Intended Audience and Reading Suggestions

Before beginning production, the document is meant for all parties involved to review and agree upon all needs and features. The sponsors, advisors, developers, and subject matter experts are examples of stakeholders. The project scope is stated in the paper, along with a thorough explanation of the features and requirements. It will make the project more understandable to all users. The paper also lists a few restrictions and limits as well as the system's functioning conditions.

2.5 Overall Description

2.5.1 Product Perspective

The product is planned to be an open-source project and is a web-based system leveraging Ethereum Blockchain technology. SIGNIFIED provides a simple mechanism for universities to easily verify their documents from issuing authorities without the need of a certain level of human intervention.

Signified allows the students to upload their digital documents which will be verified by verifying authority. After verification, documents will receive a hash value which will eventually provide access to the document. Viewers can also view the verification process between the student and verifying authority using the hash value of the document.

2.5.2 Product Functions

The major functions of Signified include:

- **User Registration:**
Allows the users to register by providing relevant information, which can then be used to interact with the application.
- **User Login:**
Logging into the system using email and password.
- **Documents Upload:**
Allows the students to upload their digital documents and fill in the details required for successfully uploading documents.
- **Document Verification:**
Allows the verifiers to verify the documents uploaded by the students based on the credibility of information provided by the students.

- **Blockchain Storage:**

After a successful verification process, the fraudulent documents will be rejected while the verified documents will be stored and a hash will be returned which is a unique identifier for each document. This document hash will be stored on Ethereum Blockchain via smart contracts.

- **Document Sharing:**

Allows the students to share the hash of the document with the viewers who can access these verified documents to check its credibility using the hash of the documents.

2.5.3 User Classes and Characteristics

The users which will interact with the system are as follows:

- **Students:**

Students will be able to upload digital documents and fill in the details required for successfully uploading documents.

- **Verifying Authority:**

Verifying authority will be the user who will verify the documents uploaded by students. They will provide originality, authenticity, and integrity to the documents. After verification, documents will have a hash value which will eventually provide access to the document.

- **Viewers:**

Viewer will be the user who can see the verification process between the student and verifying authority using the hash value of the document.

2.5.4 Operating Environment

This is a web-based system and hence will require the operating environment for a client and server. An internet connection and a browser like Mozilla Firefox, Google Chrome or Internet Explorer etc. must be available. Ethereum wallet extension should be installed. It will be executed on a Cloud-Based Platform. The cloud server will use a SQL database running on the cloud.

2.5.5 Design and Implementation Constraints

- **Technical Issues:**

This system is provisioned to provide safety and security of users' data as the students are uploading their confidential documents. Secondly, high complexity in implementing this blockchain based solution can also arise as a technical issue.

- **Operational Issues:**

Insufficient resources including time, cost and memory limitations.

2.5.6 User Documentation

User documentation components such as user manuals, on-line help, and tutorials(demo video) will be delivered along with the system. The system will have an “About” tab, which will provide basic guidelines on how to use this web application. Other tutorials and support forms will be made available in case to report any bug or other support related issues.

2.5.7 Assumptions and Dependencies

The following are the main assumptions:

- The users of the system have basic computer knowledge and skills required to handle and manage the system.
- The users of the system have an Ethereum wallet which is necessary to operate the system.
- A stable internet connection is available for the functioning of this web application.

2.6 External Interface Requirements

2.6.1 User Interfaces

The user interface for the software shall be compatible with any browser such as Internet Explorer, Mozilla or Chrome by which user can access to the system. The following tool is used for the user interface.

- Front end software: ReactJs

2.6.2 Hardware Interfaces

Since the application must run over the internet, all the hardware required to connect to the internet will be a hardware interface for the system. as for e.g., Modem, WAN – LAN, Ethernet Cross-Cable otherwise WIFI can also be used for internet connection. Also, for Windows operating systems it is designed for 32-bit or 64-bit.

2.6.3 Software Interfaces

Following are the software used for SIGNIFIED.

Operating System	Windows Mac
Database	PostgreSQL along with AWS
Backend Software	NodeJs
Blockchain Development	Ethereum Ganache Truffle

	MetaMask
--	----------

2.6.4 Communications Interfaces

The communication with the server is through HTTP. The system is accessible through all web browsers that can render React, HTML, CSS, and JavaScript pages.

2.7 System Features

This section provides the functional requirements of the product by system features and the major services provided by the product which are as follows:

2.7.1 Registration

2.7.1.1 Description and Priority

This system feature is of High priority as it allows the users to register to the system to access it.

2.7.1.2 Stimulus/Response Sequences

The user will click the sign-up button which will open the registration form.

2.7.1.3 Functional Requirements

2.7.1.3.1 The system shall allow the users to connect their Ethereum wallets with the system.

2.7.1.3.2 The system shall allow the users to register to the system by providing the following data fields:

- a) First name and Last name
- b) Email address of user
- c) Type of user (Student/Verifier)

2.7.2 Login

2.7.2.1 Description and Priority

This system feature is of High priority as it allows the users to login into the system.

2.7.2.2 Stimulus/Response Sequences

The user will click the sign in button which will open the login form.

2.7.2.3 Functional Requirements

2.7.2.3.1 The system shall allow registered users to login to the system to by providing the following data fields:

- a) Email address of user
- b) Password

2.7.2.3.2 When the system receives the user's login information the system shall:

- a) Reject it if it does not contain the registered user's:
 - 1. Email address
 - 2. Password
- b) Notify the individual about acceptance or rejection with a message.

2.7.2.3.3 The system shall navigate the user to the main page, once a user is successfully logged in.

2.7.2.3.4 The system shall terminate the logged in user session when the Sign Out button is clicked from the navigation bar.

2.7.3 User Profile

2.7.3.1 Description and Priority

This system feature is of medium priority as it allows the users to view their profile.

2.7.3.2 Stimulus/Response Sequences

The user will click the profile button from the navigation bar which will open the profile page.

2.7.3.3 Functional Requirements

2.7.3.3.1 The system shall allow registered and logged in users to open their profile page.

2.7.3.3.2 The system shall allow the users to view their profile details.

2.7.4 Upload Document

2.7.4.1 Description and Priority

This system feature is of High priority as it allows the users to upload their certificate and provide all their details.

2.7.4.2 Stimulus/Response Sequences

The system shall allow the user to click the upload button from the navigation bar which will direct them to the page.

2.7.4.3 Functional Requirements

2.7.4.3.1 The system shall allow users to fill the form required for their certificate verification.

2.7.4.3.2 The system shall allow the users to upload their certificates required for the form completion.

2.7.5 Verify Document

2.7.5.1 Description and Priority

This system feature is of High priority as it allows the verifying authority to verify the uploaded certificates.

2.7.5.2 Stimulus/Response Sequences

The system shall allow the verifier to click the verify button from their dashboard which will direct them to the respective page.

2.7.5.3 Functional Requirements

2.7.5.3.1 The system shall allow verifiers to verify the certificates that were uploaded by the students.

2.7.6 View Document

2.7.6.1 Description and Priority

This system feature is of High priority as it allows the viewers to view students' certificates by providing the document hash.

2.7.6.2 Stimulus/Response Sequences

The system shall allow the users to click the view document button from the home page which will direct them to the view page.

2.7.6.3 Functional Requirements

2.7.6.3.1 The system shall allow viewers to enter document hash to view the verified document.

2.8 Other Nonfunctional Requirements

2.8.1 Performance Requirements

2.8.1.1 The database shall be normalized to prevent redundant information and improve performance.

2.8.1.2 The product shall be web based and has to be run from a web server.

2.8.1.3 The performance shall depend upon hardware components of the client/customer.

2.8.2 Safety Requirements

2.8.2.1 Data Storage

The customer's web browser shall never display a user's password. It shall always be echoed with special characters representing typed characters

2.8.2.2 Data Transfer

The system shall not leave any cookies on the user's computer containing any of the user's confidential information.

2.8.3 Security Requirements

2.8.3.1 Security Measures

The system shall embed security measures to ensure that the records present in the system are secure and no unauthorized personnel can access them.

2.8.3.2 Secure Development

The software system defined in this SRS must follow industry recommended practices for secure software development.

2.8.4 Software Quality Attributes

2.8.4.1 Portability

The system shall be portable as it is a web-based application. The users shall be able to access it from anywhere through internet connection.

2.8.4.2 Performance

The web-based application shall have a responsive layout.

2.8.4.3 Usability

The system shall have an easy to learn interface and allow users to accomplish their goals without errors.

2.8.4.4 Maintainability

The system shall update features and fix bugs and deploy them quickly without extra downtime.

2.9 Other Requirements

2.9.1 Database Requirements

2.9.1.1 The system shall store the data in the cloud database. The stored data includes:

- a) User information for login
- b) Students' information regarding document verification application
- c) Students' documents

2.9.1.2 The system shall store the verified documents' hash on the blockchain which will be used to access the verified documents.

2.10 Use Case Diagram

Use Case Diagram of Student:

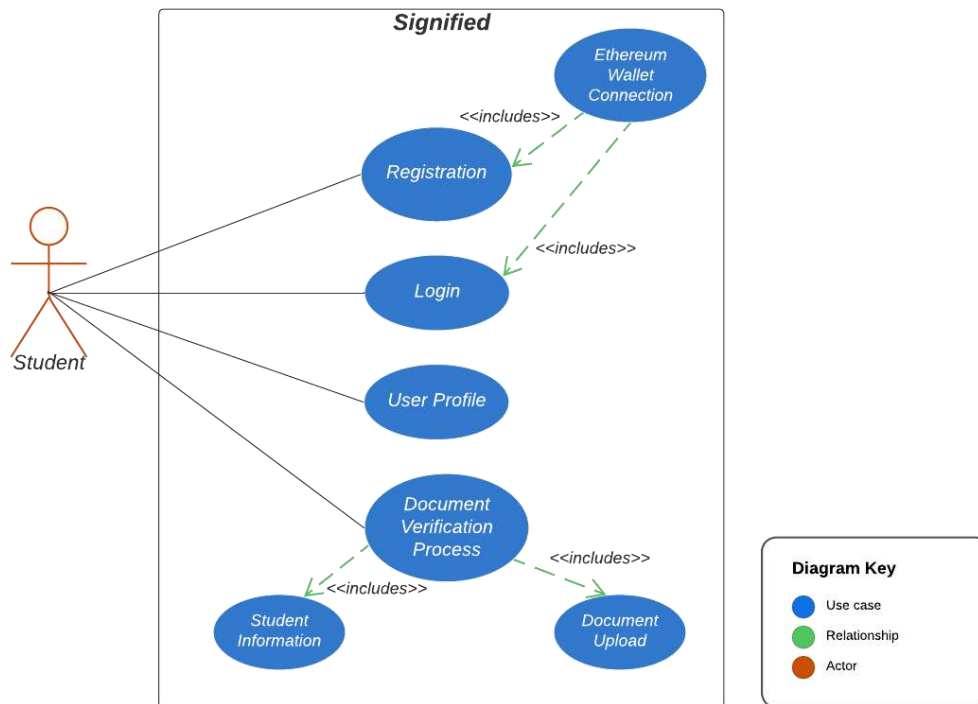


Figure 1: Use Case Diagram of Student

Use Case Diagram of Verifier:

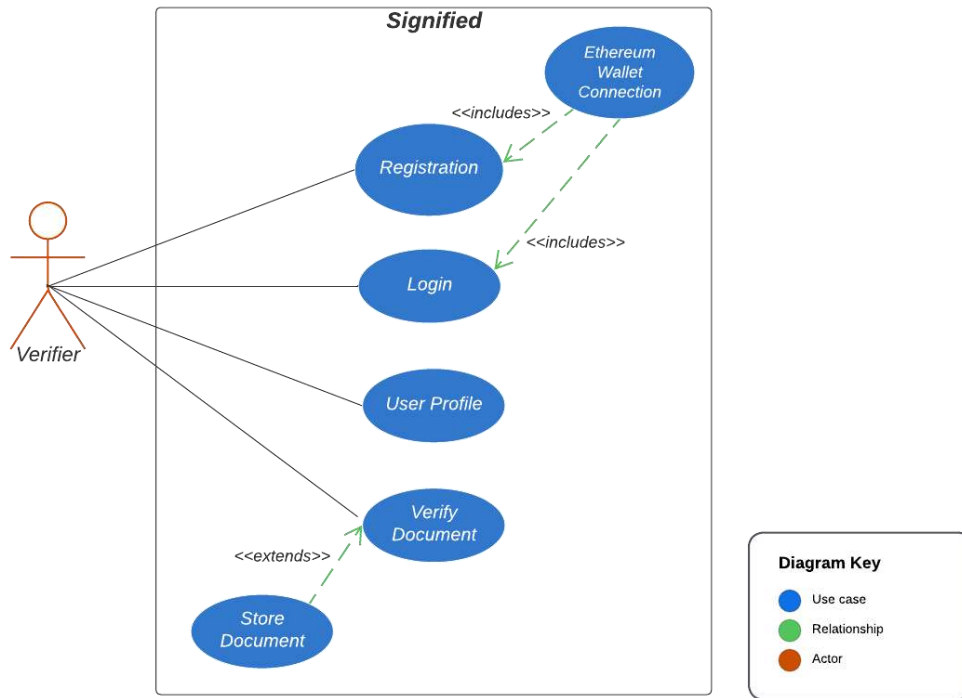


Figure 2: Use Case Diagram of Verifier

Use Case Diagram of Viewer:

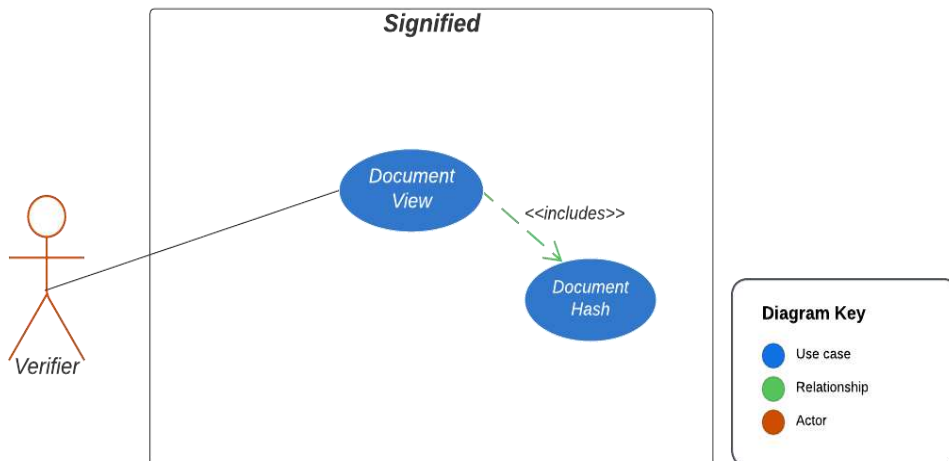


Figure 3: Use Case Diagram of Viewer

Use Case Diagram of System:

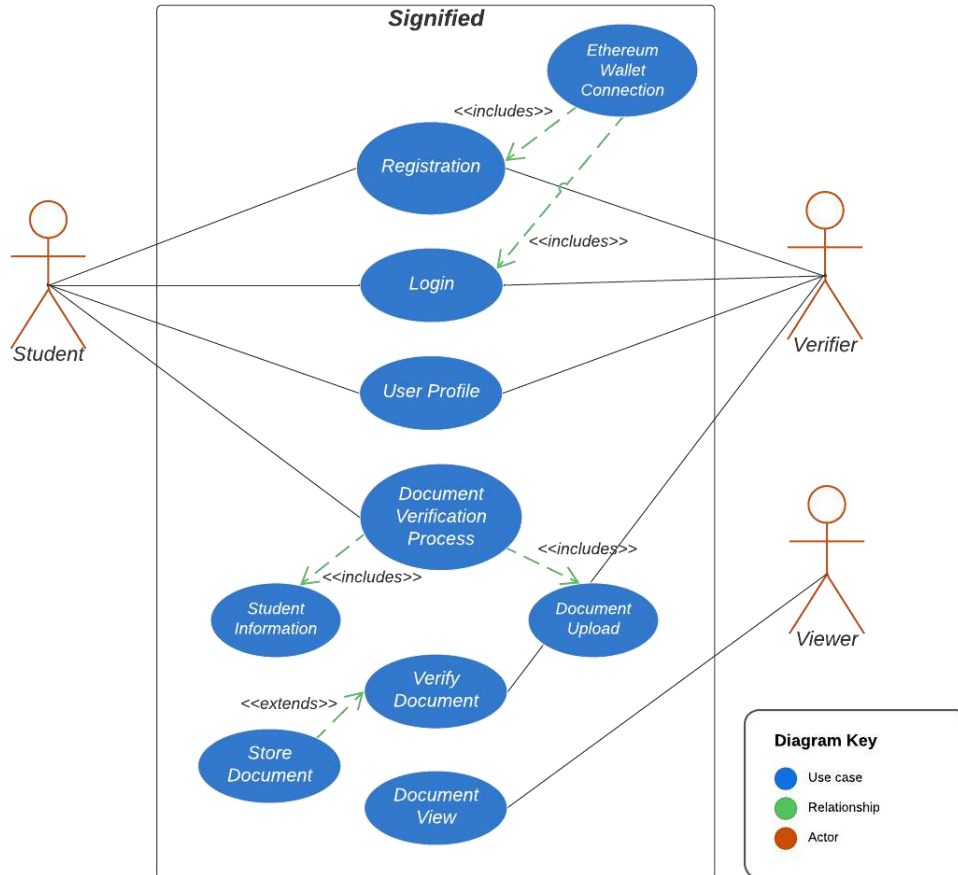


Figure 4: Use Case Diagram of System

3. Software Document Specification:

3.1 Introduction

The introduction of the Software Design Specification (SDS) provides an overview of the entire SDS with purpose, scope, definitions, acronyms, abbreviations, references and overview of

the project. The aim of this document is to gather and analyze and give an in-depth insight of the project **SIGNIFIED-Decentralized Ledger for Document Verification**. The detailed design flow of **SIGNIFIED** is also provided in this document.

3.2 Purpose

The purpose of this document is to present a detailed description of SIGNIFIED-Decentralized Ledger for Document Verification. It will basically describe functions and characteristics of the system, its interfaces, what the system will perform, the limitations that must be met for it to function, and how the system will respond to outside stimuli. This document will be submitted to the higher authorities for approval and is meant for both the stakeholders and the system developers.

3.3 Product Scope

SIGNIFIED is an automated verification system which can reduce the time required for the document verification process. We aim to develop a blockchain based solution for online documents verification. Using the Ethereum Blockchain technology, we can achieve a decentralized system to verify documents without the need of a certain level of human intervention. This will drastically improve the document verification process's efficiency along with the benefits of security, reliability and transparency and make document verification process easier, safer and more convenient for users.

The verification of documents like educational certificates is an intricate field that requires several difficult methods to validate.[1] The most crucial documents given to students by their universities are the educational certificates such as the degree and transcript.[2] Since the process of issuing the certificates is not very verifiable, fake or doctored certificates may be produced very readily. Also, these documents need to be verified by HEC for studying abroad which is a lengthy process. In this regard, blockchain technology has lately come to light as a viable method potential means for validating the document verification process as well as an important tool to deal with document fraud.

3.3 Definitions, acronyms and abbreviations

The commonly used definitions, acronyms and abbreviations in the Software Design Document is mentioned below:

- **MetaMask:** The leading self-custodial wallet. The safe and simple way to access blockchain applications and web3.
- **Infura:** Provides the tools and infrastructure that allow developers to easily take their blockchain application from testing to scaled deployment.
- **Amazon S3:** Cloud object storage with industry-leading scalability, data availability, security, and performance.
- **PostgreSQL:** Open-source object-relational database system.

Some of the abbreviations used in this document are mentioned below:

AWS	Amazon Web Services
SQL	Structured Query Language
P2P	Peer to Peer
ETH	Ether
S3	Simple Storage Sever
SRS	Software Requirements Specification
IEEE	Institute of Electrical and Electronics Engineers
HEC	Higher Education Commission
SDS	Software Design Specification

3.4 Overview of the Document

Software Design document is a comprehensive document that provides a detailed description of the system's design. It outlines the design of a system from a high-level perspective and provides a detailed description of the system's architecture, components, data flow and user interface design. The document basically includes the following sections:

- Introduction
- Architecture
- Data Flow
- Design Models
- Process Flow
- Database Schema
- User Interface Design

3.5 Design Considerations

3.5.1 Overview

The detailed high level design methodology for SIGNIFIED is discussed in design considerations. The product is planned to be an open-source project and is a web based system leveraging Ethereum Blockchain technology. SIGNIFIED provides a simple mechanism for universities to easily verify their documents from issuing authorities without the need of a certain level of human intervention.

3.5.2 Design Methodology

Design methodology provides a logical and systematic means of proceeding with the design process as well as a set of guidelines for decision-making. The design methodology provides a sequence of activities, and often uses a set of notations or diagrams such as design models which are also provided in this document below. For designing SIGNIFIED the following steps are taken:

- **Requirements Gathering:** This was already done and the details of which are mentioned are in Software Requirements Specification (SRS) which is submitted previously.
- **Architectural Style:** This involves the overall architecture the details of which are provided below.
- **Database Design:** This involves designing the database for storing and retrieving the value.
- **User Interface Design:** This involves designing the user interface for the system, including the screen, forms and interaction with the system.

3.5.3 Design Models

The major design models of the system are given below:

3.5.3.1 Class Diagram:

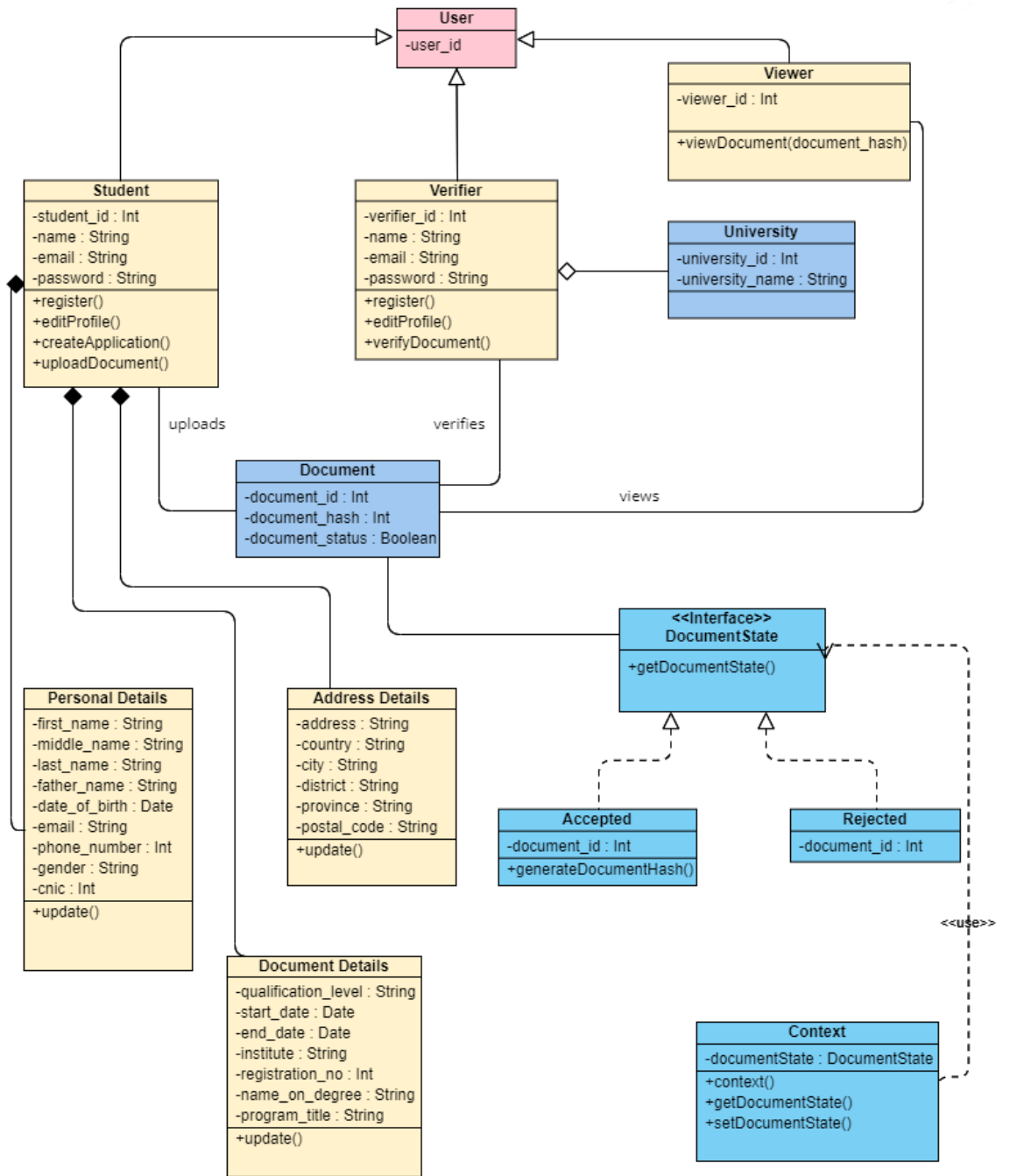


Figure 5: Class Diagram

The above class diagram describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects. The major classes include Student, Verifier, Viewer, Document and University. These classes are related to each other using association, composition and inheritance relationships. Other supporting classes are also included in this diagram.

3.5.3.2 State Transition Diagram

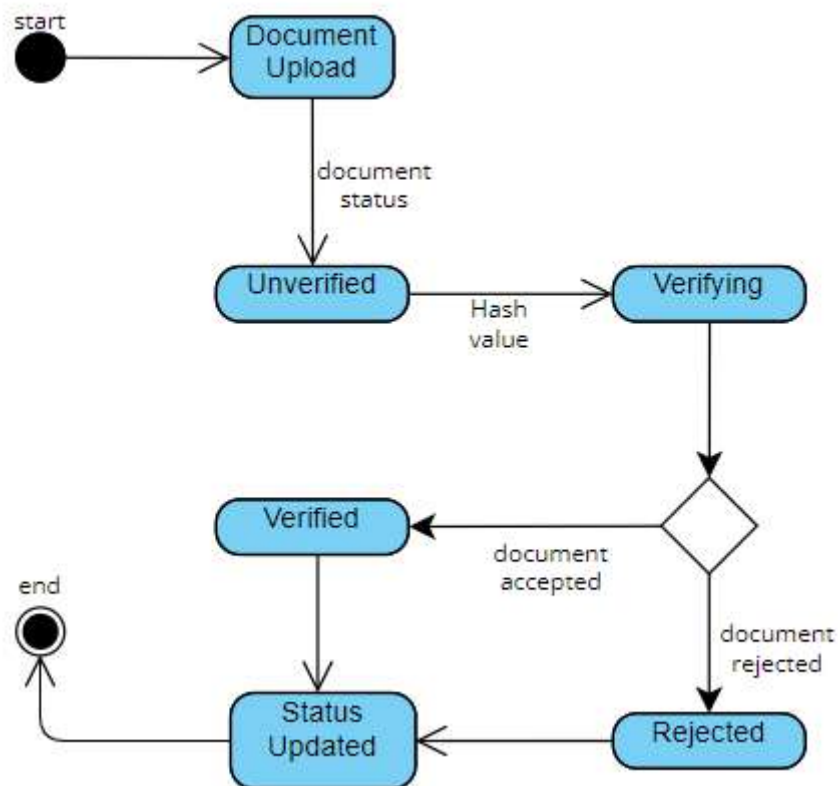


Figure 6: State Transition Diagram

There are total of 4 states in the above diagram the details of which are given below:

- **Unverified:** The document has not yet been verified or the verification process has not yet started.

- **Verifying:** The document is being verified by adding its hash value to the blockchain.
- **Verified:** The document has been verified and its hash has been added to the blockchain.
- **Rejected:** The status of the document has been changed to rejected upon verification.

3.6 System Architecture

3.6.1 Overview

By concentrating on how and why the system was decomposed in a certain way rather than on specifics of the many components, this section gives a high-level overview of the structural and functional decomposition of the system. It contains details on the system's main duties and functions.

3.6.2 System Architecture Description

SIGNIFIED is a type of distributed open-source software application that runs on a peer-to-peer blockchain network. A peer-to-peer (P2P) architecture consists of a decentralized network of peers - nodes that are both clients and servers. P2P networks distribute the workload between peers, and all peers contribute and consume resources within the network. The major components of the system include

- Front-end
- Signer
- Provider
- Ethereum Blockchain
- NodeJs Server
- PostgreSQL

- AWS S3 Bucket

The users interact with the system through the browser along with internet connection. The students and verifiers are needed to sign in through the meta mask. MetaMask is a tool that makes it easy for applications to handle key management and transaction signing. The user details are stored on the PostgreSQL database through NodeJS Server. While the documents uploaded by students for verification are stored on AWS S3 Bucket. After the verification process by the verifier, the verified documents receive document hash which is stored on Ethereum blockchain with the help of providers. These providers implement a JSON-RPC specification which ensures that there's a uniform set of methods when frontend applications want to interact with the blockchain. After successfully storing the document hash on blockchain, this verified document is fetched from blockchain when the viewer enters the specified document hash value.

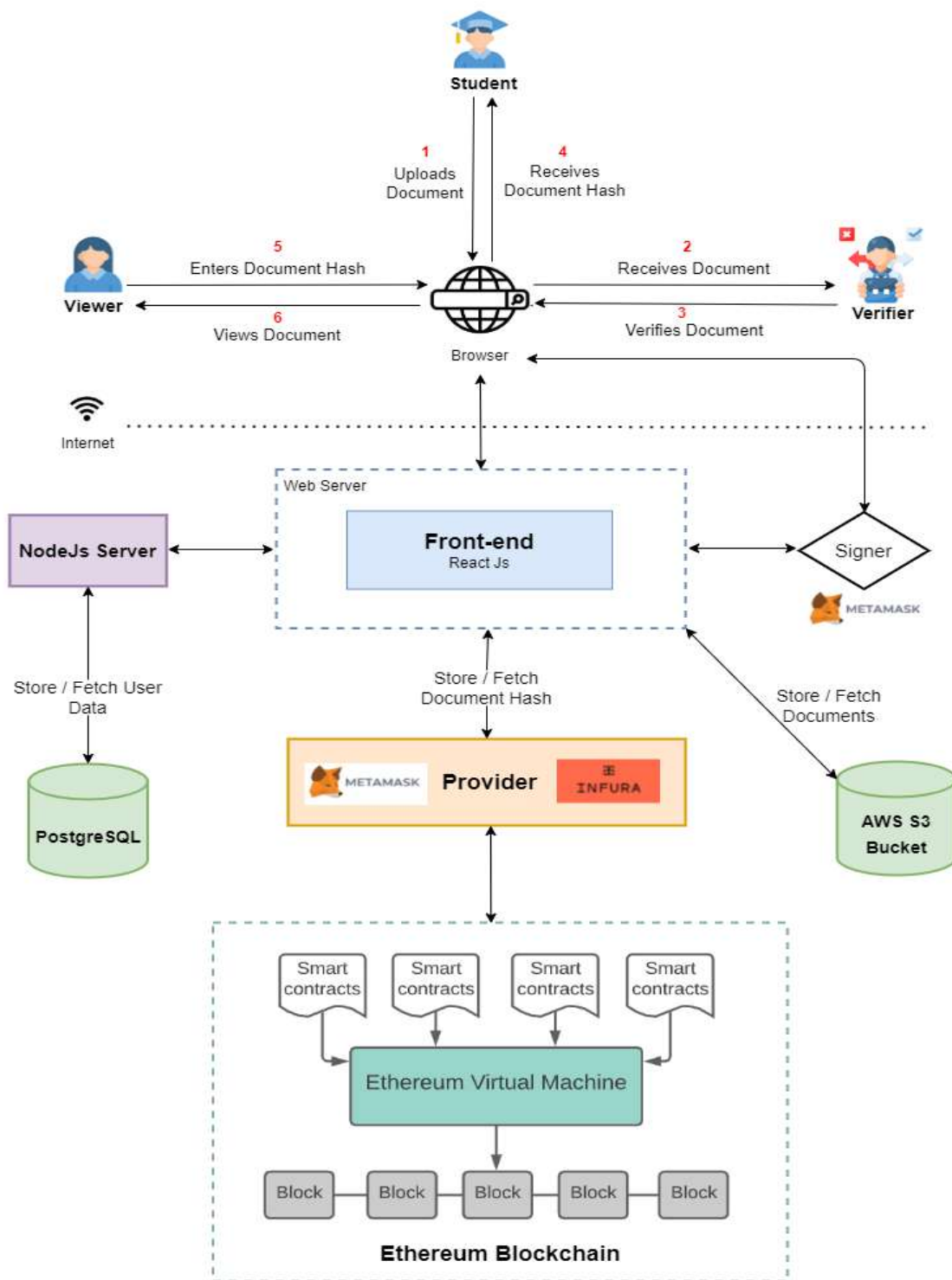


Figure 7: System Architecture

3.6.3 Process Flow

A visual overview along with an activity diagram of all the tasks and relationships involved in a process is given below.

3.6.3.1 Activity Diagram

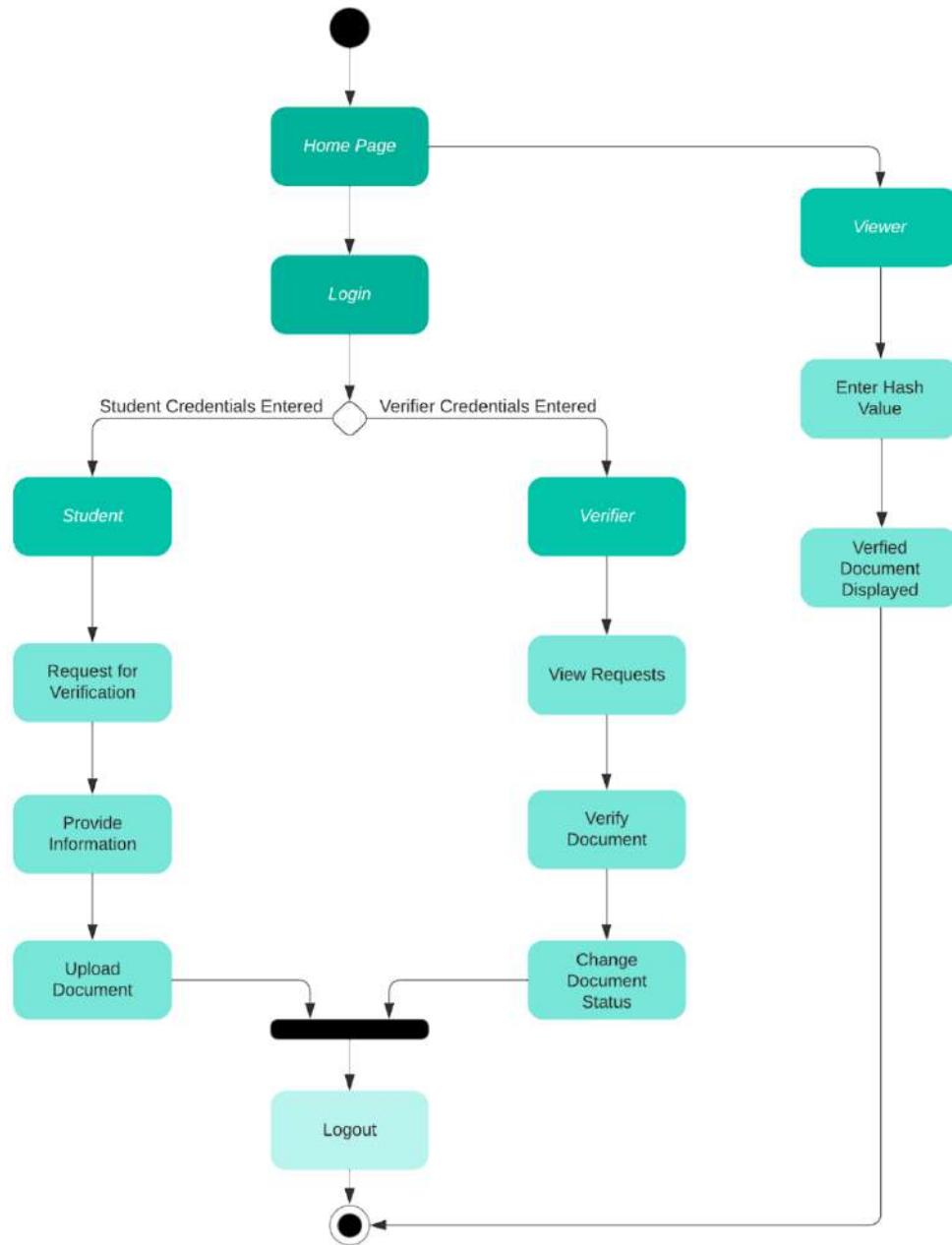


Figure 8: Activity Diagram

3.6.3.2 Description

The above activity diagram describes how activities of the system are coordinated to provide a service. It consists of different activities such as uploading documents,

verifying documents and viewing documents etc. After receiving the login credentials, condition checks are performed to check if the user is a student or a verifier. After the type of user is identified, further activities are performed and the user is logged out as the termination of the process.

3.7 Database Schema

This section provides the information related to the database schema of the project which involves tables, fields and relationships that need to be created for the design.

3.7.1 Tables, Fields and Relationships

3.7.1.1 Database

The schema name for the project is named as signified and PostgreSQL is used for creating the database schema.

3.7.1.2 New Tables

The schema of the project consists of different tables which are mentioned below:

- User
- Student
- Document
- Viewer
- Student Personal Details
- Student Address Details
- Student Document Details

<input type="checkbox"/>	Name	Owner	Partitioned table?	Comment
<input type="checkbox"/>	document	postgres	<input type="checkbox"/>	
<input type="checkbox"/>	student	postgres	<input type="checkbox"/>	
<input type="checkbox"/>	student_address_detai...	postgres	<input type="checkbox"/>	
<input type="checkbox"/>	student_document_de...	postgres	<input type="checkbox"/>	
<input type="checkbox"/>	student_personal_deta...	postgres	<input type="checkbox"/>	
<input type="checkbox"/>	university	postgres	<input type="checkbox"/>	
<input type="checkbox"/>	user	postgres	<input type="checkbox"/>	
<input type="checkbox"/>	verifier	postgres	<input type="checkbox"/>	

Figure 9: Database Tables

3.7.1.3 New Fields

Each table contains different fields and among these fields are some primary keys as well as some foreign keys.

3.8 Data Representation Diagram

3.8.1 Entity Relationship Diagram

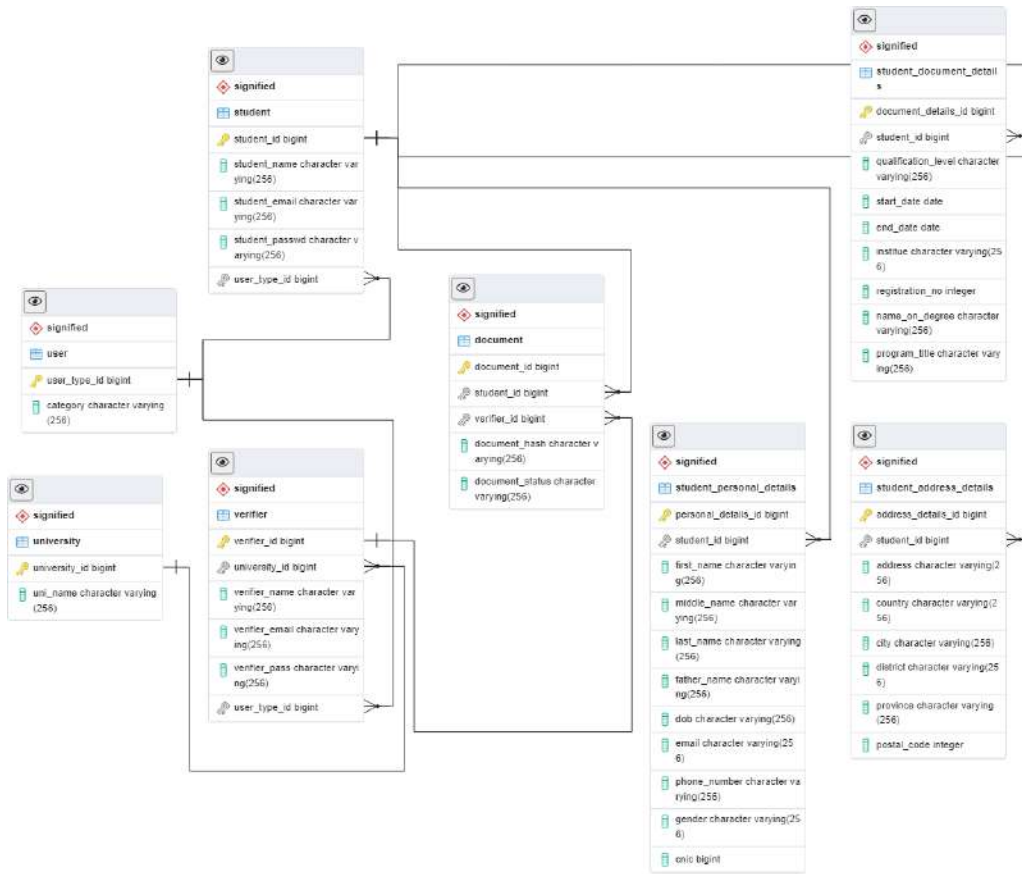


Figure 10: Entity Relationship Diagram

3.8.2 Description

The above ERD diagram is a structural representation of the schema. It consists of different entities such as user, student, verifier etc. which are basically the tables in the schema. Each table has a primary key which also acts as foreign key in some other table. For example, the primary key of the university table is university_id which has become a foreign key in the student and verifier table. University table basically consists of a unique id for each university and its name as well.

4. Software Testing

4.1 Introduction

The introduction of the Test Plan and Test Cases provides an overview of the entire document with purpose, scope, definitions, acronyms, abbreviations, references and overview of the document. The Test Plan has been created to facilitate communication within the team members. This document describes approaches and methodologies that will apply to the unit, integration and system testing of the project. It includes the objectives, testing approach, entry and exit criteria, scope and test cases of different types of testing. What will be included in and excluded from the test deliverables are both explicitly stated in this document.

4.2 Scope

The document's primary focus is on validating data in report output according to requirements specifications and GUI testing.

4.2.1 Functions to be tested

- GUI
- Data entered by users
- Search and Filters Logic
- Performance

4.2.2 Functions not to be tested

Not other than mentioned above in section 2.1

4.3 Quality Objectives

Assuring that the system satisfies all criteria, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirement, as well as maintaining the product's quality, is one of testing's core goals. The user should discover that the project has met or surpassed all of their expectations as outlined in the requirements at the conclusion of the project development cycle.

The secondary goals of testing will be to find and expose any problems and dangers connected with them, inform the project team of all known problems, and make sure that all problems are properly fixed before release. The program must be carefully and methodically tested in order to guarantee that all system components are examined and that any issues (bugs) discovered are afterwards resolved.

4.4 Test Approach

In accordance with requirements-based strategy, the method adopted is analytical, where planning, estimating, and test design are all based on an examination of the requirements specification. Through exploratory testing, test scenarios are developed. In the test strategy, every test type is identified.

4.5 Test Strategy

Testing strategy involves following major steps

- Understanding Requirements
- Preparing Test Cases
- Creating Test Data
- Executing Test Cases
- Retesting and Regression Testing
- Deployment/Delivery

4.6 Test Cases

Some of the major test cases of different types of testing of project are included in this section:

4.6.1 Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation.

Some of the major test cases are given below.

Test Case 1

Test Case ID: 1

Test Priority (Low/Medium/High): High

Module Name: Register

Test Title: Verify register with valid user

Information

Description: Test the register page

Pre-conditions: User enters valid first name, last name, email, password, university id and university name.

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
1	Navigate to register page		User should be able to register	User is navigated to	Pass
2	Provide valid first name	Anam		dashboard with successful	
3	Provide valid last name	Tariq		register	
4	Provide valid email	anamtariq@gmail.com			
5	Provide valid password	anam12345			
8	Provide valid University Id	6657			
9	Provide valid university name	Air University			
4	Click on Register button				

Test Case 2

Test Case ID: 2

Test Priority (Low/Medium/High): Medium

Module Name: Register

Test Title: Verify register with empty first name

Description: Test the register page

Pre-conditions: User enters empty first name and valid last name, email, password, university id and university name.

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
	Navigate to register page		User should be not be able to register	User is not navigated to dashboard	Pass
	Provide empty first name			dashboard with successful	
3	Provide valid last name	Tariq		register	
4	Provide valid email	anamtariq@gmail.com			
5	Provide valid password	anam12345			
8	Provide valid University Id	6657			
9	Provide valid university name	Air University			
4	Click on Register button				

Test Case 3

Test Case ID: 3

Test Priority (Low/Medium/High): Medium

Module Name: Register

Test Title: Verify register with invalid email

Description: Test the register page

Pre-conditions: User enters invalid email and valid last name, email, password, university id and university name.

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
	Navigate to register page		User should be not be able to register	User is not navigated to dashboard	Pass
	Provide empty first name	Anam		dashboard with successful	
3	Provide valid last name	Tariq		register	
4	Provide valid email	anamtariq.com			
5	Provide valid password	anam12345			
8	Provide valid University Id	6657			
9	Provide valid university name	Air University			
4	Click on Register button				

Test Case 4

Test Case ID: 4

Test Priority (Low/Medium/High): Medium

Module Name: Register

Test Title: Verify register with password length < 8

Description: Test the register page

Pre-conditions: User enters invalid password with length less than 9 and valid first name, last name, email, university id and university name.

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
	Navigate to register page		User should be not be able to register	User is not navigated to dashboard	Pass
	Provide empty first name	Anam		dashboard with successful	
3	Provide valid last name	Tariq		register	
4	Provide valid email	anamtariq@gmail.com			
5	Provide valid password	an			
8	Provide valid University Id	6657			

9	Provide valid university name	Air University			
4	Click on Register button				

Test Case 5

Test Case ID: 5

Test Priority (Low/Medium/High): High

Module Name: Register

Test Title: Verify register with email already registered

Description: Test the register page

Pre-conditions: User enters email already registered and valid first name, last name, password, university id and university name.

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
	Navigate to register page		User should be not be able to register	User is not navigated to dashboard	Pass

	Provide empty first name	Anam		dashboard with successful	
3	Provide valid last name	Tariq		register	
4	Provide valid email	anamtariq@gmail.com			
5	Provide valid password	Anam123			
8	Provide valid University Id	6657			
9	Provide valid university name	Air University			
4	Click on Register button				

Test Case 6

Test Case ID: 6

Test Priority (Low/Medium/High): High

Module Name: Login

Test Title: Verify login with valid email and password

Description: Test the login page

Pre-conditions: User has valid email and password

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to login page		User should be able to login	User is navigated to	Pass
2	Provide valid email	Email=hamna@gmail.com		dashboard with successful	
3	Provide valid password	Password: hamna12345		login	
4	Click on Login button				

Test Case 7

Test Case ID: 7

Test Priority (Low/Medium/High): High

Module Name: Login

Test Title: Verify login with valid email and wrong pass

Description: Test the login page

Pre-conditions: User has valid email and invalid password

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to login page		User should be able to login	User is not navigated to	Pass
2	Provide valid username	Email= hamna@gmail.com		dashboard with unsuccessful	
3	Provide invalid password	Password: hamna123		login	
4	Click on Login button				

Test Case 8

Test Case ID: 8

Test Priority (Low/Medium/High): High

Module Name: Login

Test Title: Verify login with wrong email and missing pass

Description: Test the login page

Pre-conditions: User has valid wrong email and missing pass

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
1	Navigate to login page		User should be able to login	User is not navigated to	Pass
2	Provide invalid username	Email= hamnagmail.com		dashboard with unsuccessful	
3	Provide password (empty field)	Password: empty		login	
4	Click on Login button				

Test Case 9

Test Case ID: 9

Test Priority (Low/Medium/High): High

Module Name: Login

Test Title: Verify login with wrong email syntax

Description: Test the login page

Pre-conditions: User has wrong email syntax

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
------	------------	-----------	-----------------	---------------	-------

1	Navigate to login page		User should be able to login	User is not navigated to	Pass
2	Provide valid username	Email= hamna@gmailcom		dashboard with unsuccessful	
3	No password given	Password: empty		login	
4	Click on Login button				

Test Case 10

Test Case ID: 10

Test Priority (Low/Medium/High): High

Module Name: Login

Test Title: Verify login with empty email and wrong password

Description: Test the login page

Pre-conditions: User has empty email and wrong password

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	
1	Navigate to login page				Pass

			User should be able to login	User is not navigated to	
2	Provide username (empty field)	Email=empty		dashboard with unsuccessful	
3	Provide invalid password	Password: 1234		login	
4	Click on Login button				

Test Case 11

Test Case ID: 11

Test Priority (Low/Medium/High): High

Module Name: New Application

Test Title: Enter all valid Information

Description: Test the personal details page

Pre-conditions: User enters valid first name, last name, email, contact number, father name, cnic and address details.

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
	Navigate to personal details page		User should be able to go to next page	User is navigated to document details page	Pass

	Provide valid first name	Saira			
3	Provide valid last name	Sarwar			
4	Provide valid email	saira@gmail.com			
5	Provide valid contact	03150157071			
8	Provide valid Father Name	Sawar			
9	Provide valid Address	Lahore			
4	Click on Next button				

Test Case 12

Test Case ID: 12

Test Priority (Low/Medium/High): High

Module Name: New Application

Test Title: Enter all invalid or empty Information

Description: Test the personal details page

Pre-conditions: User enters invalid or empty first name, last name, email, contact number, father name, cnic and address details.

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
	Navigate to personal details page		User should not be able to go to next page	User is not navigated to document details page	Pass
	Provide empty first name	Saira			
3	Provide valid last name	Sarwar			
4	Provide invalid email	saira.com			
5	Provide valid contact	03150157071			
8	Provide valid Father Name	Sawar			
9	Provide valid Address	Lahore			
4	Click on Next button				

Test Case 13

Test Case ID: 13

Test Priority (Low/Medium/High): High

Module Name: New Application

Test Title: Enter all valid Information

Description: Test the document details page

Pre-conditions: User enters valid qualification level, start date, end date, university and program title.

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
	Navigate to personal details page		User should be able to go to next page	User is navigated to document upload page	Pass
	Provide valid qualification level	Bachelors			
3	Provide valid start date	09/09/2019			
4	Provide valid end date	03/06/2023			
5	Provide valid university	NUST			
8	Provide valid Program Title	BE Software Engineering			
4	Click on Next button				

Test Case 14

Test Case ID: 14

Test Priority (Low/Medium/High): High

Module Name: New Application

Test Title: Enter invalid or empty Information

Description: Test the document details page

Pre-conditions: User enters valid qualification level, start date, end date, university and program title.

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
	Navigate to personal details page		User should not be able to go to next page	User is not navigated to document upload page	Pass
	Provide empty qualification level				
3	Provide invalid start date	0028			
4	Provide valid end date	03/06/2023			
5	Provide valid university	NUST			
8	Provide valid Program Title	BE Software Engineering			
4	Click on Next button				

Test Case 15

Test Case ID: 15

Test Priority (Low/Medium/High): High

Module Name: New Application

Test Title: Enter valid document

Description: Test the document upload page

Pre-conditions: User enters valid document.

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
	Navigate to personal details page		User should be able to go to next page	User is navigated to preview page	Pass
	Provide valid File type	Document.jpg			
4	Click on Next button				

Test Case 16

Test Case ID: 16

Test Priority (Low/Medium/High): High

Module Name: New Application

Test Title: Enter invalid document

Description: Test the document upload page

Pre-conditions: User enters valid document.

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
	Navigate to personal details page		User should not be able to go to next page	User is not navigated to preview page	Pass
	Provide invalid File type	Document.js			
4	Click on Next button				

Test Case 17

Test Case ID: 17

Test Priority (Low/Medium/High): High

Module Name: New Application

Test Title: Verify entered Information

Description: Test the preview page

Pre-conditions: User verifies entered Information.

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Notes
	Navigate to preview page		User should be able to submit application	User is navigated to dashboard	Pass
	Checks all Information displayed				
4	Click on Next button				

```
PASS src/App.test.js
  LoginComponent
    ✓ renders LoginComponent without crashing (144 ms)
    ✓ renders the component (126 ms)
    ✓ initializes with login mode (43 ms)
    ✓ toggles between login and signup modes (223 ms)
    ✓ renders Navbar component (65 ms)
    ✓ initial mode is set correctly (47 ms)
    ✓ mode toggles correctly (99 ms)
    ✓ displays sign-up message in signup mode (55 ms)
    ✓ mode toggles correctly (115 ms)
    ✓ toggle message changes when mode toggles (119 ms)
    ✓ Toggle checkbox updates the mode in LoginComponent (65 ms)
    ✓ app div has correct class name based on mode (33 ms)
    ✓ toggle checkbox toggles the mode (163 ms)
    ✓ form-block div has correct class name based on mode (33 ms)
    ✓ form-block-wrapper div has correct class name based on mode (57 ms)

Test Suites: 1 passed, 1 total
Tests:       15 passed, 15 total
Snapshots:   0 total
Time:        5.947 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more. □
```

Figure 11: Unit Testing

4.6.2 System Testing

System testing basically evaluates how the various components of an application interact together in the full, integrated system or application. System testing can be done using Selenium.

Test Case 1

Test Plan ID	TP_FUNCT_USER-SIGNUP
Description	Checks that the user is registered first.
Test Plan Priority	High
Requirement Type	Functional
Pre-Conditions	User is not registered before.
Post Conditions	User information is visible in website database.
Testing Strategy	Test using Selenium IDE or any other suitable IDE.
Expected Result	User (student or verifier) is successfully registered and his/her information is visible in website database.

Test Case 2

Test Plan ID	TP_FUNCT_CUSTOMER-LOGIN
Description	Checks that the user is logged in either as a student or verifier.
Test Plan Priority	High
Requirement Type	Functional
Pre-Conditions	User is registered.
Post Conditions	User can successfully access his account.
Testing Strategy	Test using Selenium IDE or any other suitable IDE.
Expected Result	User can successfully access his account.

Test Case 3

Test Plan ID	TP_FUNCT_LOGOUT
Description	Checks that the user is logged out of his/her account.
Test Plan Priority	High
Requirement Type	Functional
Pre-Conditions	User is not registered before.
Post Conditions	User information is visible in website database.
Testing Strategy	Test using Selenium IDE or any other suitable IDE.
Expected Result	User is successfully logged out of their accounts but their data remains saved in the database.

Test Case 4

Test Plan ID	TP_FUNCT_LOGOUT
Description	Checks that the student can all his/her applications
Test Plan Priority	High
Requirement Type	Functional
Pre-Conditions	Student is logged In.
Post Conditions	User information is visible in website database.
Testing Strategy	Test using Selenium IDE or any other suitable IDE.
Expected Result	Student can successfully view his/her applications saved in the database.

Test Case 5

Test Plan ID	TP_FUNCT_LOGOUT
Description	Checks that the student can create new application
Test Plan Priority	High
Requirement Type	Functional
Pre-Conditions	Student is logged In.
Post Conditions	New application is created.
Testing Strategy	Test using Selenium IDE or any other suitable IDE.
Expected Result	Student can successfully create his/her application and it is saved in the database.

Test Case 6

Test Plan ID	TP_FUNCT_LOGOUT
Description	Checks that the verifier can view all students' applications
Test Plan Priority	High
Requirement Type	Functional
Pre-Conditions	Verifier is logged In.
Post Conditions	User information is visible in website database.
Testing Strategy	Test using Selenium IDE or any other suitable IDE.
Expected Result	Verifier can successfully view students' applications saved in the database.

Test Case 7

Test Plan ID	TP_FUNCT_LOGOUT
Description	Checks that the verifier can accept/reject students' applications
Test Plan Priority	High
Requirement Type	Functional
Pre-Conditions	Verifier is logged In.
Post Conditions	Application is accepted or rejected.
Testing Strategy	Test using Selenium IDE or any other suitable IDE.
Expected Result	Verifier can successfully accept/reject students' applications and the status is updated in the database.

4.6.3 Performance Testing

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size. We can do performance testing using Jmeter by increasing the number of users and their requests to check the performance of the project.

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	11:12:15.253	Thread Group 1...	HTTP Request	8	✓	2085	164	8	2
2	11:12:15.754	Thread Group 1...	HTTP Request	6	✓	2085	164	6	3
3	11:12:16.254	Thread Group 1...	HTTP Request	7	✓	2085	164	7	1
4	11:12:16.753	Thread Group 1...	HTTP Request	5	✓	2085	164	5	1
5	11:12:17.253	Thread Group 1...	HTTP Request	3	✓	2085	164	3	1
6	11:12:17.754	Thread Group 1...	HTTP Request	16	✓	2085	164	16	5
7	11:12:18.251	Thread Group 1...	HTTP Request	3	✓	2085	164	3	1
8	11:12:18.750	Thread Group 1...	HTTP Request	4	✓	2085	164	4	1
9	11:12:19.261	Thread Group 1...	HTTP Request	15	✓	2085	164	15	8
10	11:12:19.751	Thread Group 1...	HTTP Request	5	✓	2085	164	5	1
11	11:12:20.250	Thread Group 1...	HTTP Request	5	✓	2085	164	5	2
12	11:12:20.760	Thread Group 1...	HTTP Request	12	✓	2085	164	12	4
13	11:12:21.261	Thread Group 1...	HTTP Request	8	✓	2085	164	8	2
14	11:12:21.752	Thread Group 1...	HTTP Request	9	✓	2085	164	9	3
15	11:12:22.257	Thread Group 1...	HTTP Request	3	✓	2085	164	3	1
16	11:12:22.747	Thread Group 1...	HTTP Request	3	✓	2085	164	3	1
17	11:12:23.246	Thread Group 1...	HTTP Request	5	✓	2085	164	5	3
18	11:12:23.745	Thread Group 1...	HTTP Request	3	✓	2085	164	3	1
19	11:12:24.245	Thread Group 1...	HTTP Request	4	✓	2085	164	4	1
20	11:12:24.744	Thread Group 1...	HTTP Request	3	✓	2085	164	3	0

Scroll automatically?
 Child samples?
No of Samples: 20
Latest Sample: 3
Average: 6
Deviation: 3

Figure 12: Performance Testing

5. Implementation and Deployment

This section includes deployment details and system integration details such as cloud services and tools used in the system.

5.1 Tools and Software Used

5.1.1 Truffle

Truffle is a development framework for blockchain-based applications. It is designed to make the process of developing, testing, and deploying smart contracts on a blockchain network easier and more efficient. Truffle provides a suite of tools and libraries that developers can use to manage the entire software development lifecycle of a blockchain application, from writing and testing smart contracts to deploying them on the blockchain network. It supports popular blockchain platforms like Ethereum, Hyperledger Fabric, and Corda.



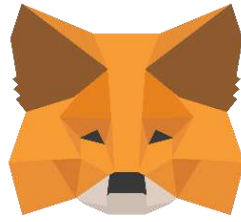
5.1.2 Ganache

Ganache is a personal blockchain network that can be used for testing and development purposes. It allows developers to simulate a blockchain network on their local machine, making it easier and faster to develop and test blockchain applications without needing to deploy on a public network.



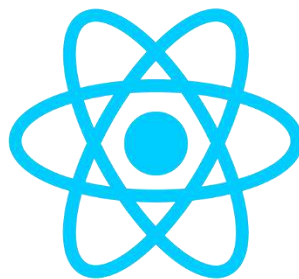
5.1.3 MetaMask

MetaMask is a browser extension and mobile app that allows users to interact with decentralized applications (dApps) on the Ethereum blockchain. It serves as a digital wallet and a bridge between a user's browser and the Ethereum network, making it easier to manage Ethereum-based assets and access dApps.



5.1.4 React Js

React.js is an open-source JavaScript library used for building user interfaces (UIs). React.js allows developers to build UI components that are modular, reusable, and easy to maintain. It uses a declarative syntax that makes it easier to describe and update the UI based on changes to the application's state or data. React.js also includes a virtual DOM (Document Object Model) that can efficiently update only the parts of the UI that have changed, which can result in faster performance and a better user experience.



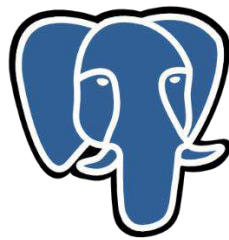
5.1.5 Node Js

Node.js is an open-source server-side JavaScript runtime environment that enables developers to build scalable and high-performance applications. It uses an event-driven, non-blocking I/O model, which makes it well-suited for building real-time, data-intensive applications that can handle a large number of concurrent connections.



5.1.6 PostgreSQL

PostgreSQL (short for Postgres) is an open-source relational database management system (RDBMS) that uses and extends the SQL language. It is designed to be highly scalable, secure, and extensible, and is used by many large and small organizations for a wide range of applications.



PostgreSQL

5.1.7 AWS S3 Bucket

Amazon S3 (Simple Storage Service) is a cloud-based storage service provided by Amazon Web Services (AWS) that allows users to store and retrieve data from anywhere on the web. An S3 bucket is a container for storing objects (files) that can be accessed from anywhere on the web.



5.1.8 Deployment on S3 Bucket

The documents which the student uploads are stored in Amazon AWS S3 bucket as shown below:

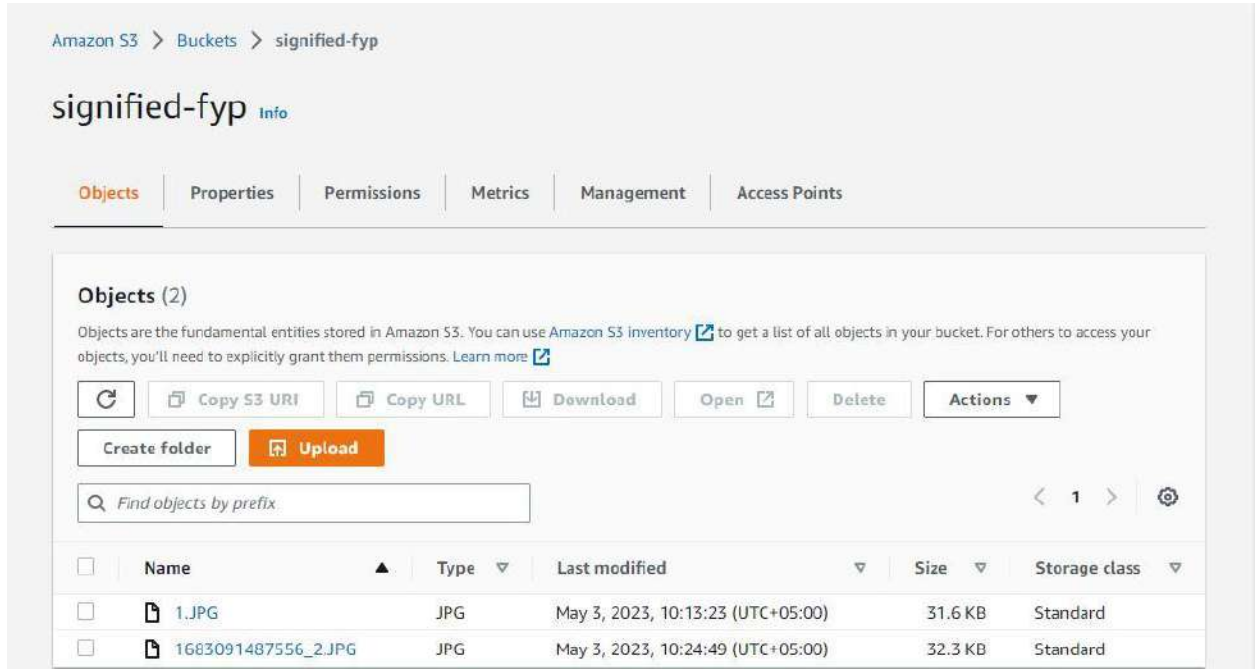


Figure 13: Documents Uploaded on AWS S3 Bucket

6. User Interface Design

6.1 Home Page

The home page is the default or front page of the system. It is the first page that visitors see when they view the system. It consists of sections such as about, contact and major information regarding the usage of the system.

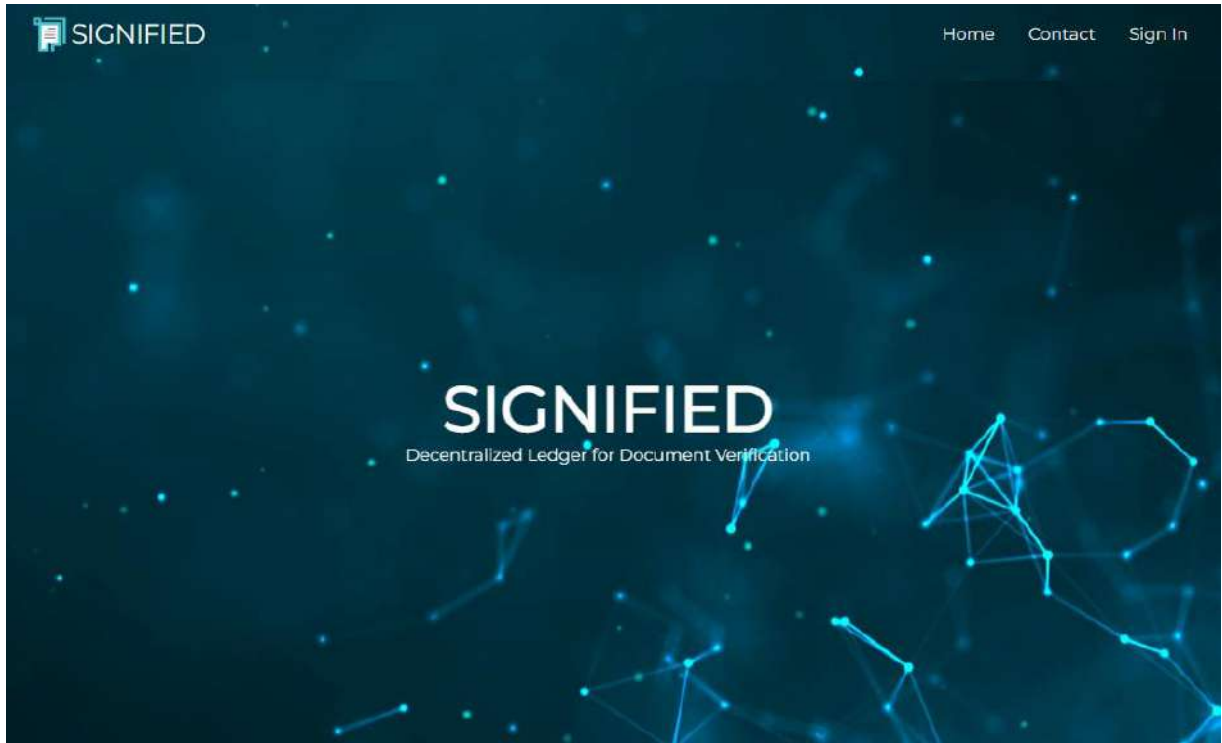


Figure 14: Home page

6.2 Sign Up Page

For the first time using the system, the users will be required to register by entering his/her name, email address and password using the Signup Page. Upon successful registration, the dashboard will be displayed.

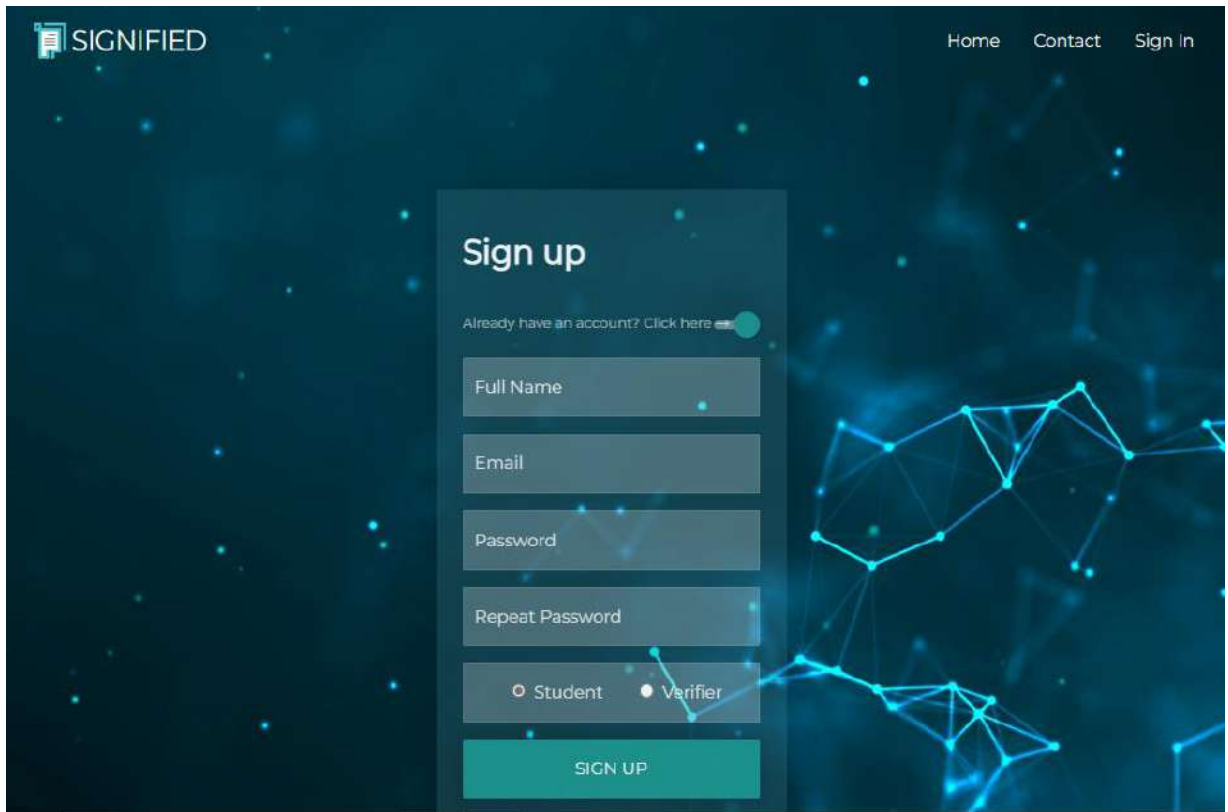


Figure 15: Sign up page

6.3 Login Page

The registered users will be required to login by entering his/her email address and password using the Sign in Page. Upon successfully signing in, the dashboard will be displayed.

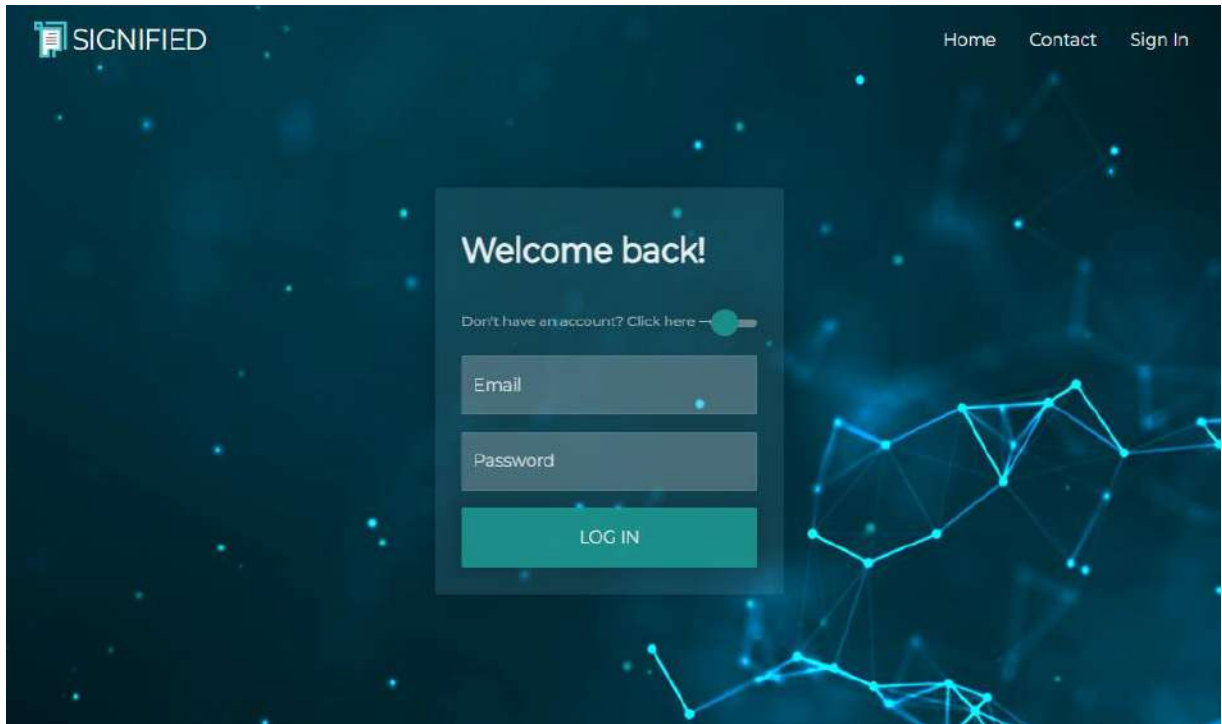


Figure 16: Login page

6.4 Student Dashboard Page

The Dashboard is an important part of the system, it basically handles all the users' actual interaction with the system. The user will be able to view their account and applications' information and status.

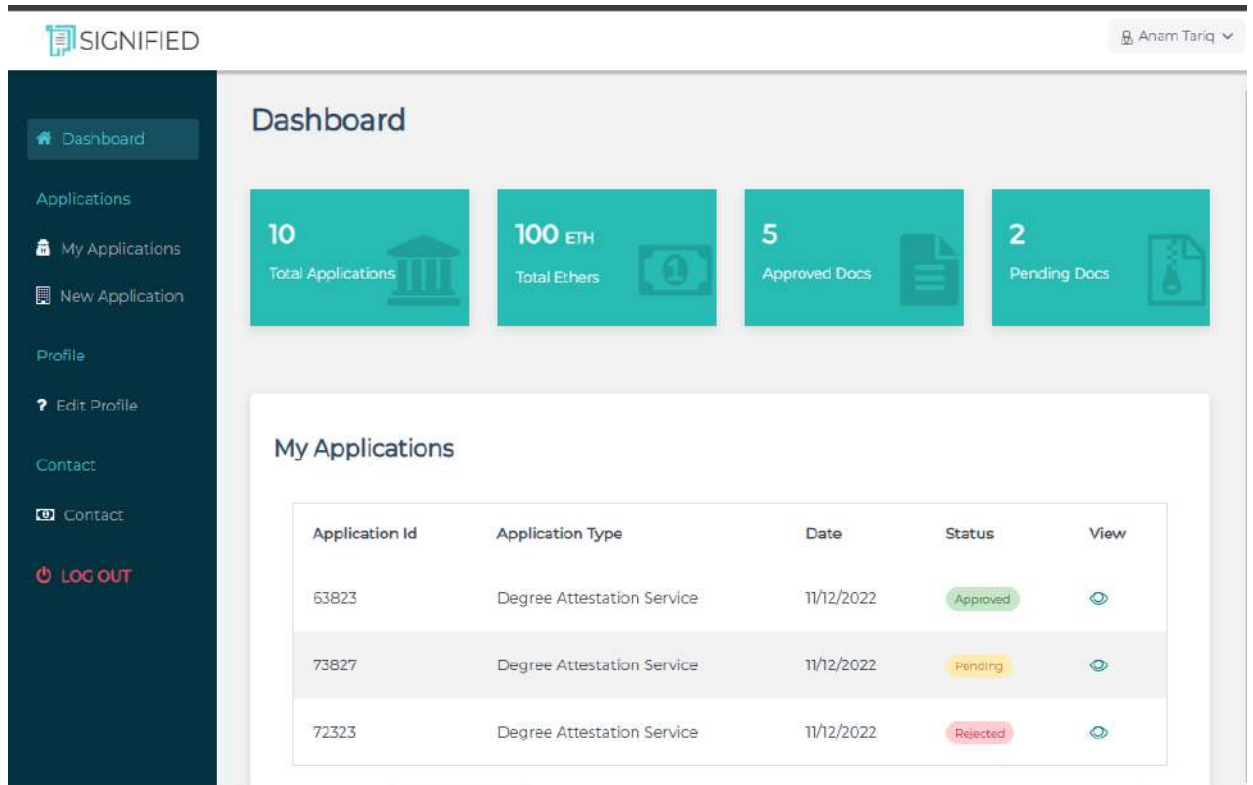


Figure 17: Student Dashboard page

6.5 New Application Page

The New Application Page allows the user to create a new application for document verification. It consists of five major sections required for successful submission of details and documents.

6.5.1 Personal Details

This section allows the user to enter their personal details such as First Name, Last Name, Email, Phone Number, Date of Birth, Gender and CNIC as well as address details consisting of Address, City, Province and Postal Code.

SIGNIFIED Anam Tariq

New Application

1 Personal Details
 2 Detail of Degree
 3 Document Upload
 4 Preview Application
 5 Submission

Personal Details

Personal Details

First Name* <input type="text" value="Enter Your First Name"/>	Middle Name <input type="text" value="Enter Your Middle Name"/>	Last Name* <input type="text" value="Enter Your Last Name"/>
Father Name <input type="text" value="Enter Your Father Name"/>	Date of Birth* <input type="text" value="mm/dd/yyyy"/>	Email* <input type="text" value="Enter Your Email"/>
Mobile Number* <input type="text" value="03xx-xxxxxxx"/>	Gender* <input type="text" value="Select Gender"/>	CNIC* <input type="text" value="xxxxx-xxxxxx-x"/>

Address Details

Figure 18: Personal Details

6.5.2 Document Details

This section allows the user to enter the qualification details such as Qualification Level, Start Date and End Date with degree/certificate details such as Roll No, Name on Degree and Program Title as well as Degree/Certificate Awarding Institute Details which includes Institute Name.

SIGNIFIED Anam Tariq

New Application

- 1 Personal Details
- 2 Detail of Degree
- 3 Document Upload
- 4 Preview Application
- 5 Submission

Document Details

Qualification Details

Qualification Level*

Start Date*

End Date*

Degree/Certificate Details

Registration/Roll No*

Name on Degree*

Program Title on Degree*

Degree/Certificate Awarding Institute Details

Degree Awarding Institute*

Figure 19: Document Details

6.5.3 Document Upload

This section allows the user to upload the document which needs to be verified.

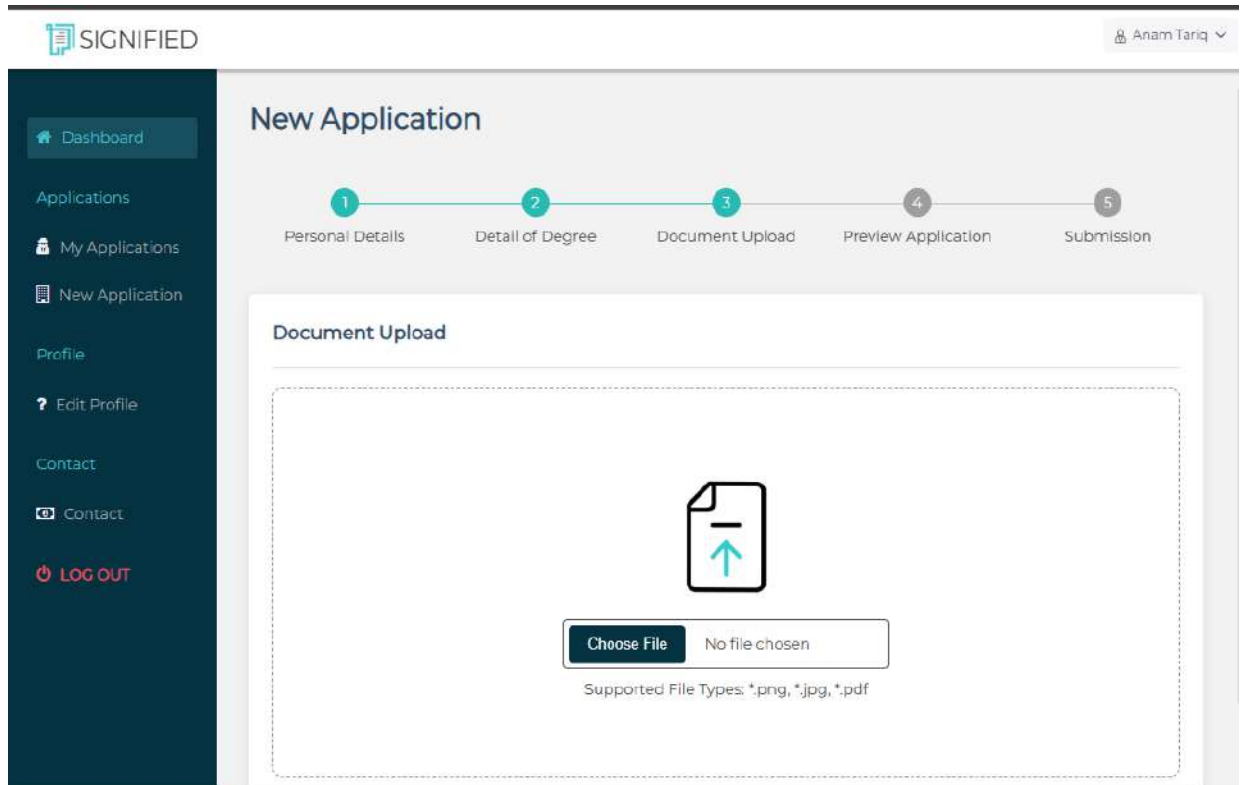


Figure 20: Document Upload

6.5.4 Preview Details

This section allows the user to view all the details entered previously before submitting the application for verification purpose.

SIGNIFIED Anam Tariq

New Application

- 1 Personal Details
- 2 Detail of Degree
- 3 Document Upload
- 4 Preview Application
- 5 Submission

Preview Application

Personal Details

First Name: Last Name: Date of Birth:

Email: Mobile Number: Gender:

CNIC:

SIGNIFIED Anam Tariq

Qualification Details

Qualification Level: Start Date: End Date:

Degree/Certificate Awarding Institute Details

Degree Awarding Institute:

Degree/Certificate Details

Registration/Roll No: Name on Degree: Program Title on Degree:

Document Attached:

Figure 21: Preview Details

6.5.5 Submission

After properly viewing the details, the application is successfully submitted for verification.

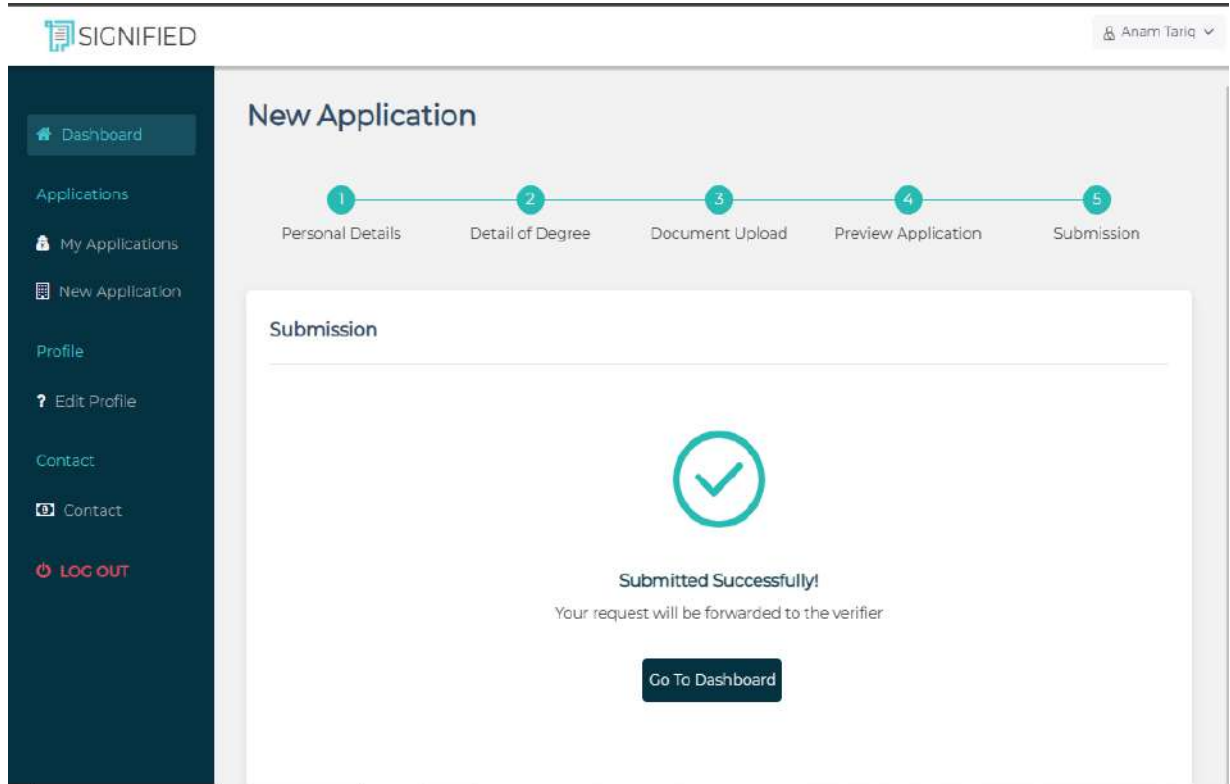


Figure 22: Submission

6.6 View Application Page

The view application page allows the student to view their submitted application. They can view applications' information and status.

SIGNIFIED Laiba Ansar

Degree Attestation Application

Application id: 5030

Personal Details

First Name Laiba	Last Name Ansar	Date of Birth 2006-12-31
Email laiba@gmail.com	Mobile Number 0315-0157071	Gender Female
CNIC 61101-8427092-2		

Address Details

Address House No. C23, Street 2, Officers Colony	Country Pakistan	City Wah Cantonment
--	----------------------------	-------------------------------

SIGNIFIED Laiba Ansar

Degree Rewarding Institute
National University of Sciences & Technology (NUST)

Degree/Certificate Details

Registration/Roll No 576576	Name on Degree Laiba Ansar	Program Title on Degree BESE
---------------------------------------	--------------------------------------	--

Document Attached
Capture.PNG

Document Hash
LaibaAnsar_BESE_6alb1e71

Document Deep Link
[LaibaAnsar_BESE_6alb1e71](#)

Document QR Code


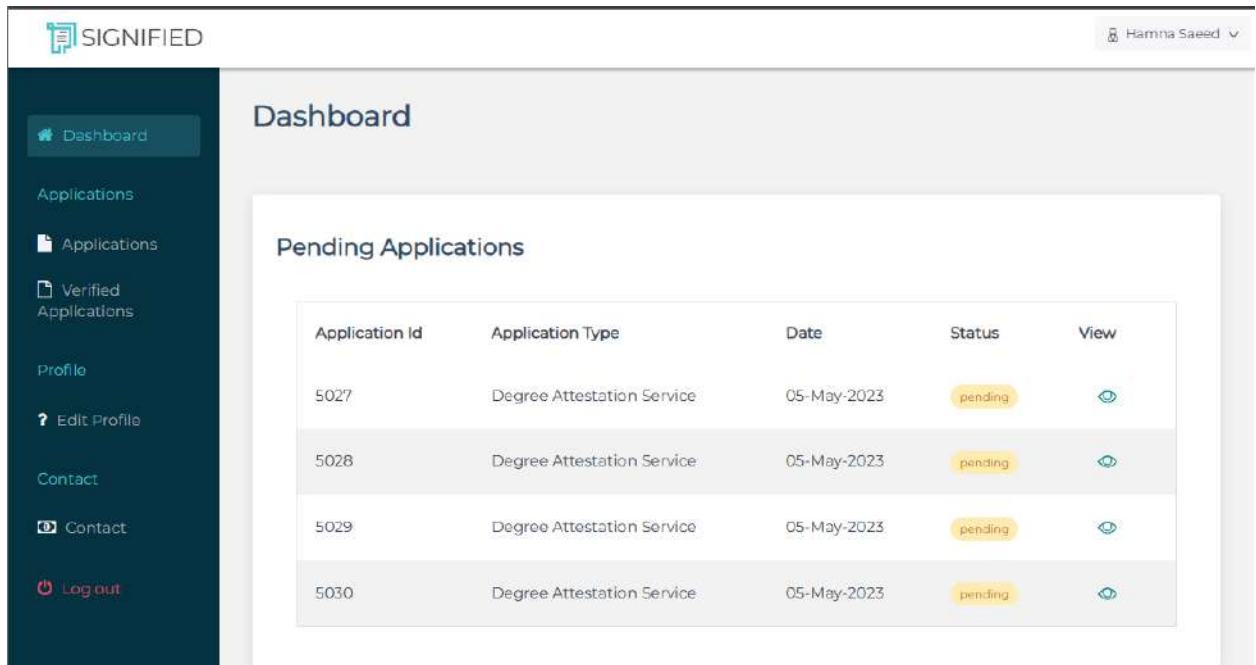


Figure 23: View Application Page

6.7 Verifier Dashboard Page

Verifier dashboard page is displayed when a user logs into the system as a verifier. This page allows the verifier to view the pending applications submit by the students.



The screenshot displays the Verifier Dashboard Page. At the top left, the logo 'SIGNIFIED' is visible. In the top right corner, the user's name 'Hamna Saeed' is shown with a dropdown arrow. The dashboard has a dark teal sidebar on the left with the following menu items: Dashboard (selected), Applications, Verified Applications, Profile, Edit Profile, Contact, and Log out. The main content area is titled 'Dashboard' and features a 'Pending Applications' section. This section contains a table with the following data:

Application Id	Application Type	Date	Status	View
5027	Degree Attestation Service	05-May-2023	pending	
5028	Degree Attestation Service	05-May-2023	pending	
5029	Degree Attestation Service	05-May-2023	pending	
5030	Degree Attestation Service	05-May-2023	pending	

Figure 24: Verifier Dashboard Page

6.8 Verified Applications Page

This page allows the verifier to view the accepted or rejected applications after verification.

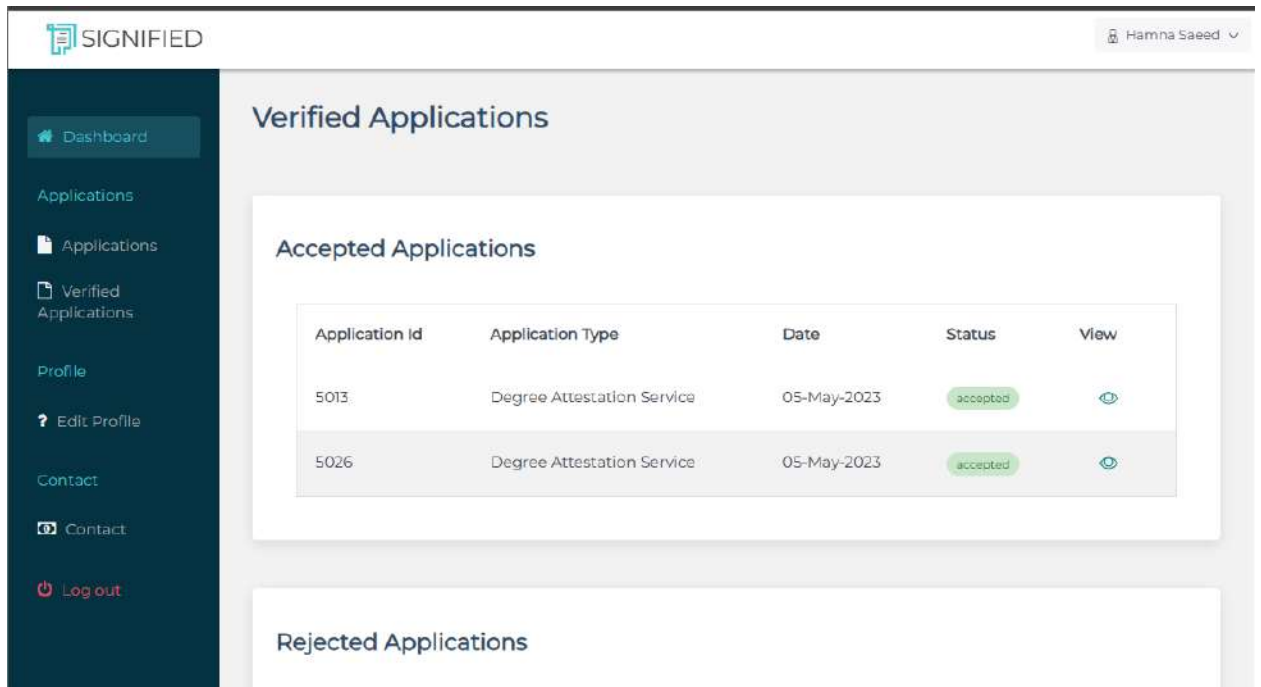


Figure 25: Verified Applications Page

6.9 Verifier View Application

Verifier can view application and can accept or reject this application.

SIGNIFIED Hamna Saeed

Degree Attestation Application

Application id: 5027

Personal Details

First Name Zunaira	Last Name Ali	Date of Birth 2006-12-31
Email zunaira@gmail.com	Mobile Number 0315-0157071	Gender Female
CNIC 61101-8427092-2		

Address Details

Address House No. C23, Street 2, Officers Colony	Country Pakistan	City Wah Cantonment
--	----------------------------	-------------------------------

SIGNIFIED Hamna Saeed

Details of Document 1:

Qualification Details

Qualification Level Bachelors	Start Date 02-Apr-2023	End Date 30-Apr-2023
---	----------------------------------	--------------------------------

Degree Rewarding Institute
National University of Sciences & Technology (NUST)

Degree/Certificate Details

Registration/Roll No 65765765	Name on Degree Zunaira Ali	Program Title on Degree BESE
---	--------------------------------------	--

Document Attached
Document

Figure 26: Verifier View Application

6.10 View Document through Hash Value

Viewer can view the accepted applications by entering the Hash Value of the document shared by the students.

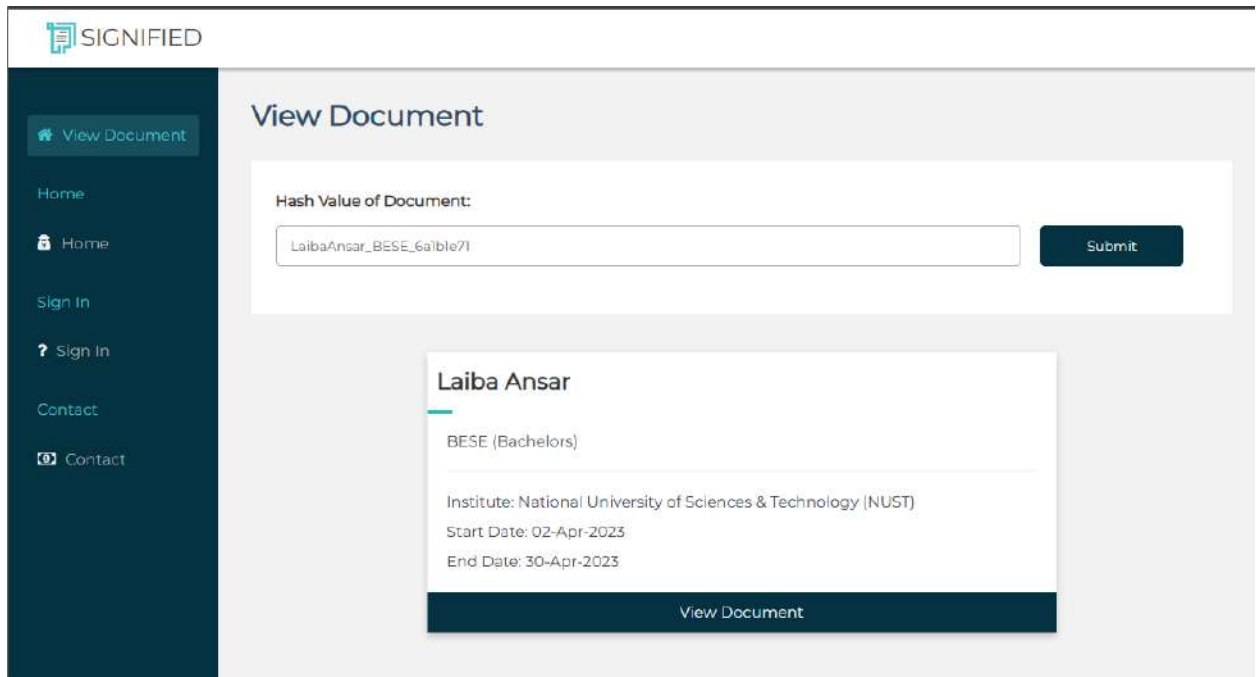


Figure 27: View Document through Hash Value

6.11 View Document though Deep Link or QR Code

Viewer can view the accepted applications through deep link or QRCode shared by the students.

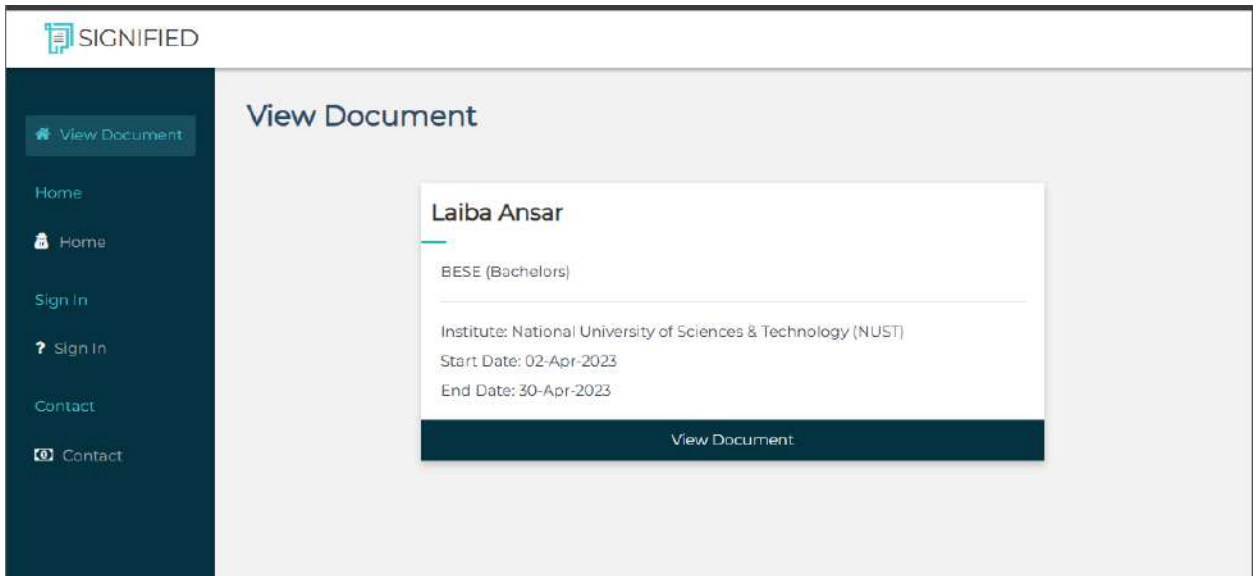


Figure 28: View Document through Deep Link or QR Code

Appendix A: Glossary

SQL	Structured Query Language
HTTP	Hyper Text Transfer Protocol
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
GUI	Graphical User Interface
HEC	Higher Education Commission
IEEE	Institute of Electrical and Electronics Engineers
AWS	Amazon Web Services
WAN	Wide Area Network

LAN	Local Area Network
WIFI	Wireless Fidelity

References

- [1] R. M. P. Q. Z. K.-K. R. C. Emmanuel Nyaletey, "'BlockIPFS-Blockchain-enabled Interplanetary File System for Forensic and Trusted Data Traceability'", in *IEEE International Conference on Blockchain*, 2019.
- [2] P. K. Maharshi Shah, "'Tamper Proof Birth Certificate Using Blockchain," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 5S3, 2019.
- [3] I. I. A. S. H. L. D. a. S. N. Meirobie, "Framework Authentication e-document using Blockchain Technology on the Government system."
- [4] T. H. K. U. o. S. a. Technology, "HKUST," 2022. [Online]. Available: [Hkust.edu.hk](http://hkust.edu.hk).
- [5] "MyDiplome," 2022. [Online]. Available: Cimea.diplo-me.eu.
- [6] CoinGeek, "University of Sharjah utilizes BSV blockchain to verify academic certificates," 2022. [Online]. Available: <https://coingeek.com/university-of-sharjah-utilizes-bsv-blockchain-to-verify-academic-certificates>.
- [7] M. M. Lab, "Blockcerts," MIT Media Lab Blockcerts and the Digital Certificates Project, 2018. [Online]. Available: <https://www.blockcerts.org>.