CSU22013/CSU33013: Group 10
Requirements Document
Blockchain publishing system

Propylon

By Anastasiya Bogoslovskaya,
Steven Cataluna,
Charles Christiansson,
Mohamed Difallah,
Alice Doherty,
Conor Doherty,
Alexander Sepelenco

Contents

1 Introduction

1.1 Overview and Purpose of System

Validation of information on the internet has gotten harder and harder with time.

The internet is filled with misinformation, untrustworthy history and suspicious sources. The need for a way of validating information found on the internet in a way that is trustworthy has become ever so important in our day and age. This is where Blockchain comes in, it will provide an effective way of validating history, current information and prevent tampering with information on the internet. The Blockchain will work with solving these problems and the website itself will give a user-friendly experience to the user, showing them visually, previous versions of information they seek as well as giving a visual indicator of the validity of the information they seek and that tampering for certain did not occur.

1.2 Scope

The following items are in scope for this project:

- Allow users to publish content, with versions being stored on the blockchain.
- The system must be able to validate the authenticity of information that is on the blockchain.
- The system must prevent tampering of information.
- The system must allow users to visually see the following:
  - Authenticity of information.
  - Previous history versions of that information, as well as who contributed to those versions.
  - If information has been tampered with.
- The system can support a pre-existing blockchain or a new one.
- The system can be built on pre-existing open source CMS such as WordPress or a personal website.
- The system allows for documents to be viewed publically.

The following items are out of scope for this project:

- The system does not need to support mobile devices, desktop is sufficient.
- The system does not need to support multiple information methods of showing something to a user, a blog will suffice.
- The system focuses on the blockchain aspect and so, a website that allows users to login and register and creating passwords and login, is out of the scope. A simple login that demonstrates the blockchain capability is sufficient.

1.3 Objectives and Success Criteria

After completing the project, a successful project will have the following criteria:

- The system will allow a user to enter the website.
- The system will show a list of documents (e.g. blog posts) a user can click on.
- The system will provide the user with information such as history, authenticity, and if it has been tampered with.
- The system will have a simple GUI for the user to have an enjoyable experience.
- The system will provide all information that is accessible to any user that wishes to see it.
- The system will prevent tampering of information.

1.4 Definitions and Abbreviations

- **UI**: User Interface; How a user interacts with a computer
- **GUI**: Graphical User Interface; A form of UI that allows the user to interact with the computer through the use of graphics.
- **Blockchain**: Interlinked blocks that contain cryptographic hashes of the previous blocks, timestamps, and transaction data.
- **Genesis block**: The start of a blockchain, it does not have a hash to point to a previous block.
- **Crypto**: Anything related to mathematical cryptography for use of secure communication.
- **CMS**: Content Management System; Software used to manage digital content.

1.5 References

- https://www.educba.com/what-is-gui
- https://en.wikipedia.org/wiki/Cryptography
- https://www.youtube.com/watch?v=wHTcrmhskto
- https://www.youtube.com/watch?v=bBC-nXj3Ng4
- https://www.youtube.com/watch?v=SSo_EIwHSd4
- https://propylon.com/platform-architecture
- www.citizensinformation.ie

2 Current System

Our client, Propylon, does not currently use blockchain technology in their current system. However, they do use specialised software which acts almost like a blockchain. There are currently many blockchains already in existence which would be suited to our project such as the Ethereum blockchain and the Polygon blockchain.

Using an existing blockchain such as the Polygon blockchain has many benefits. These include:

- Security and reliability
- Relatively cheap transaction fees
- Established documentation for developers
- Developers can work on their test net for free

Alternatively, our team could make our own blockchain. However, this would require much more effort as secure blockchains are often difficult to develop and also cost resources to maintain.

3 Proposed System

3.1 Overview

A web-based application for authoring and publishing content (e.g a blog) onto a blockchain with current and past versions of the content persisting on the blockchain.

The application should:

- Allow users of the system to see the current version of the content.
- Validate the authenticity of the content by comparing it to a version stored on the blockchain.
- View previous versions of content published to the blockchain.
- Provide a verifiable audit trail of the blockchain for all versions of content to prove that content hasn't been tampered with, changed, or otherwise modified externally

3.2 Functional Requirements

The main requirement for this project is to create a program that allows the user to publish documents to a blockchain to be validated.

Using a blockchain, our program should be able to:

- Show the audit trail of the document, ie. show the history of changes made, and the individuals who made them.
- Store all previous versions of the document, so users can view past versions that have been uploaded.
- Check and validate the document, ie. check whether the stated authors and editors actually wrote the content they're credited for.
- Display these documents publicly. For example, legal legislations and regulations proposed by the government.

3.3 Non-functional Requirements

We aim for our programme to:

- Have a basic security authentication system ie. login and password.
- Have fast performance.
- Have a user interface that is easy to navigate.

3.4 System Prototype (Models)

3.4.1 User Interface Mockups

Below is the page that the user would see after logging in. It displays basic information about the user and previews some relevant posts. The header that leads to various parts of the website is persistent throughout all pages.

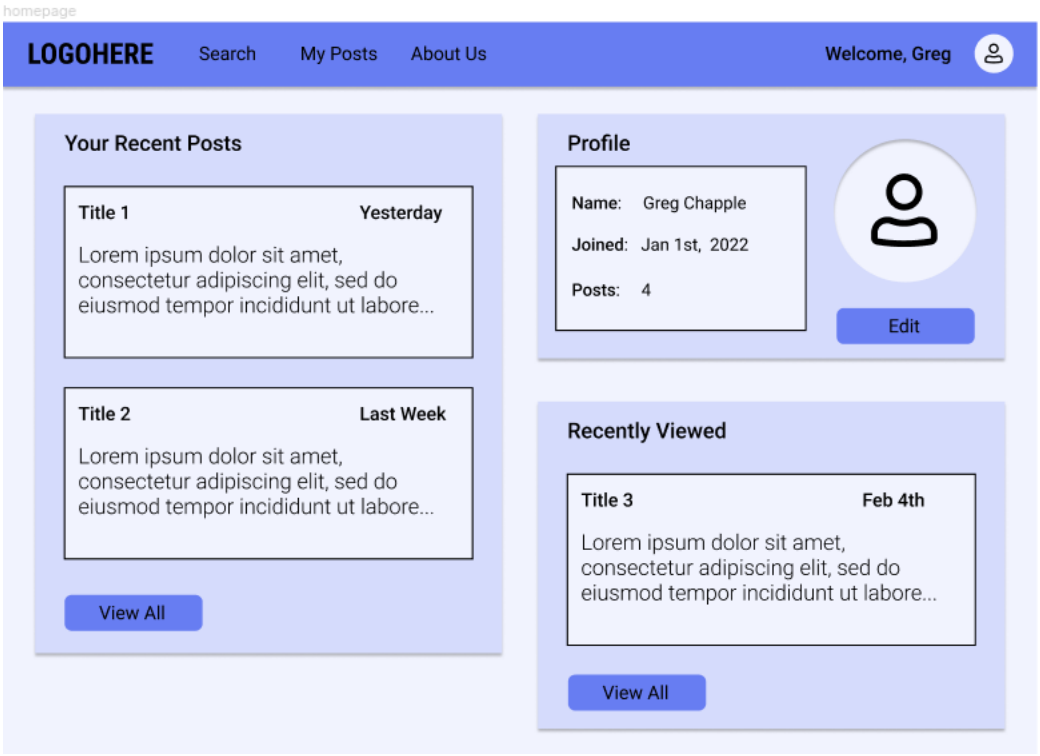

Figure 1: Homepage of site mockup

Here the user is shown searching for posts by a certain user. The page displays all posts published by that user with their title, data and current version.

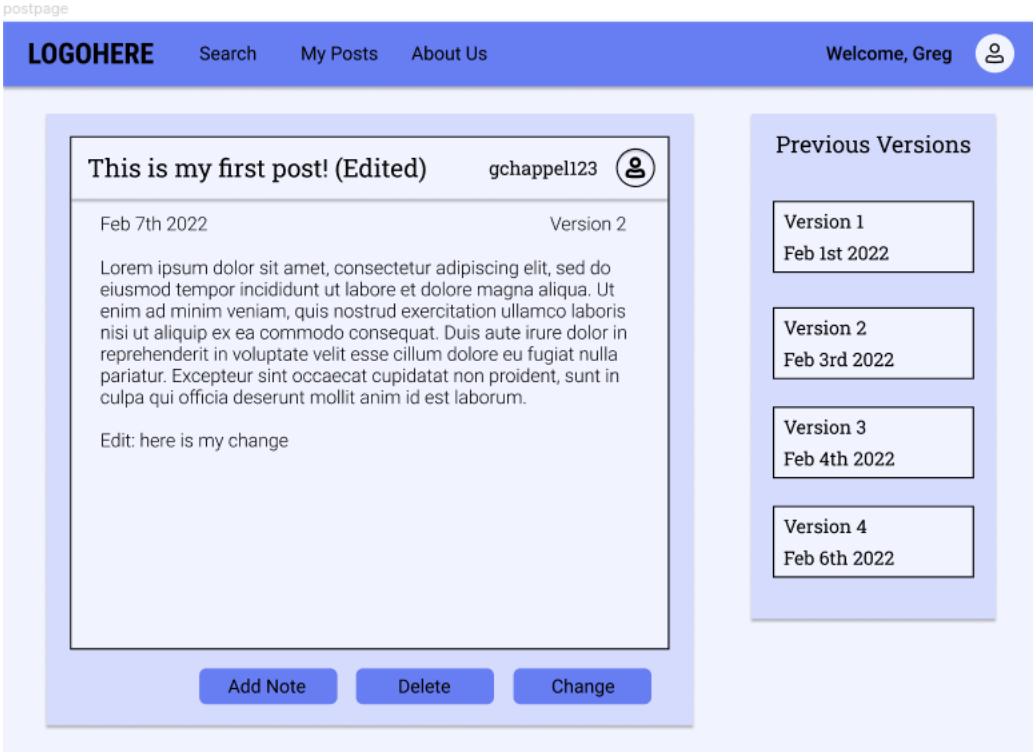

Figure 2: Postpage of site mockup

When clicking on a post from the search or the user's own list of posts, it takes the user to a page showing the entire post in its most recent version, the buttons on the right change the version of the post currently displayed.

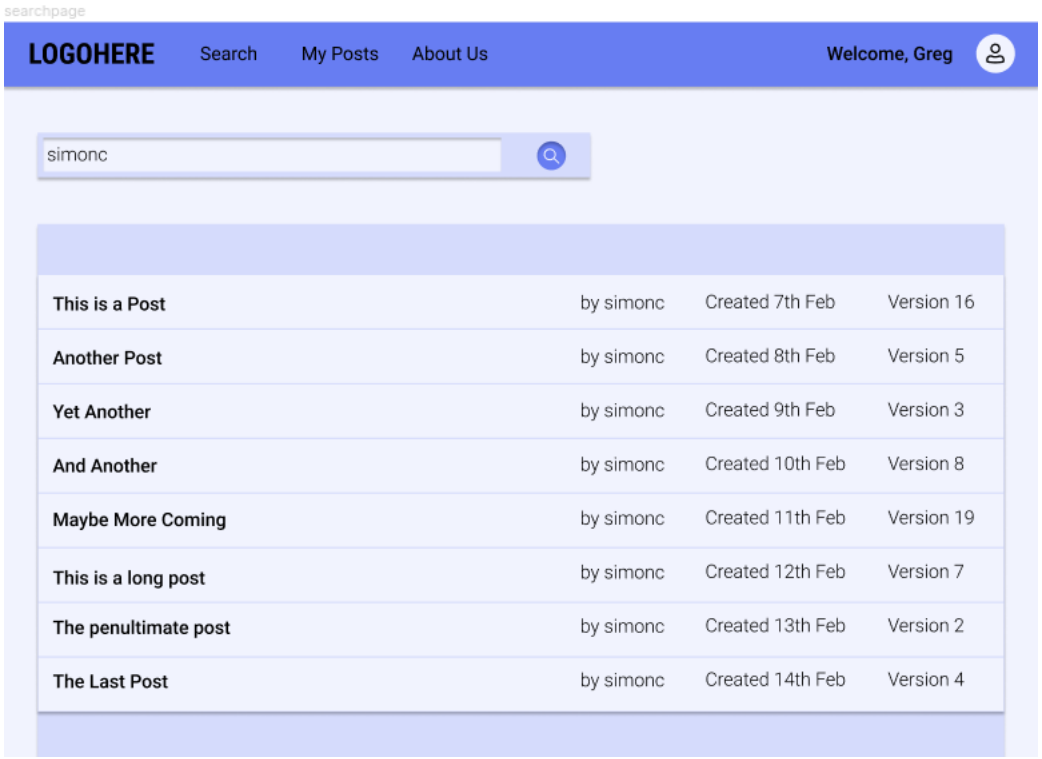

Figure 3: Seachpage of site mockup

Below this is a list of the changes made to the post, the date the change was made and a short note explaining the change made.

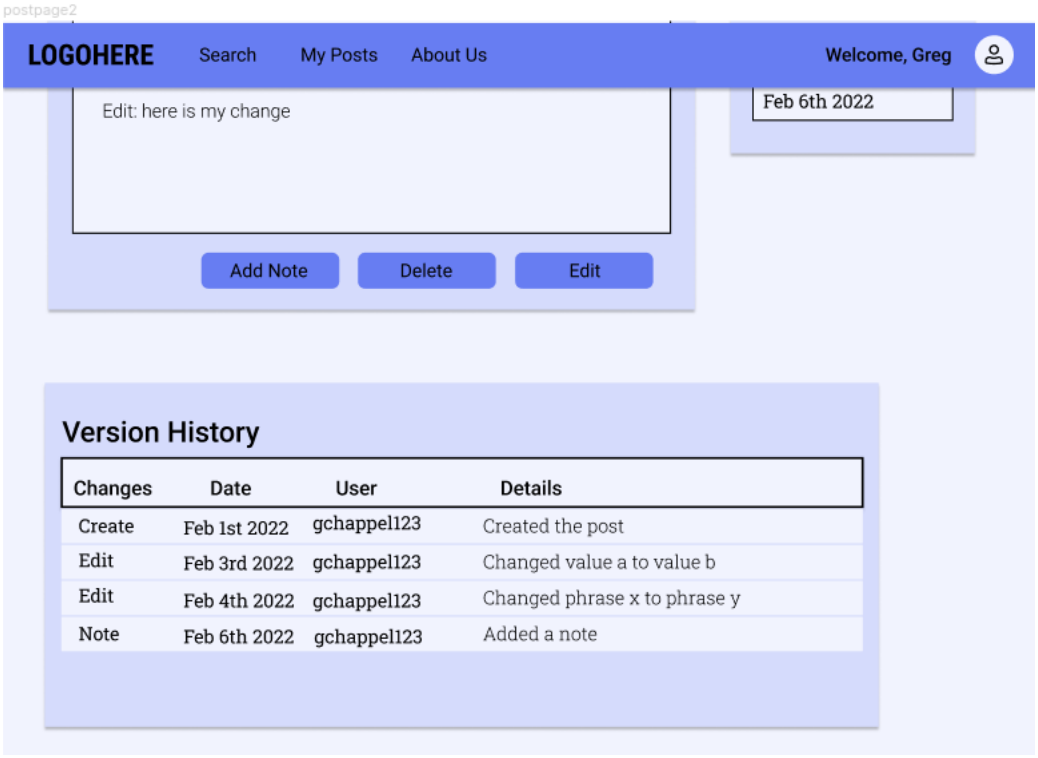

Figure 4: Postepage 2 of site mockup

3.4.2 Use Cases (including text narratives)

The following use case diagram makes the assumption that all data stored on the blockchain is valid and authentic. The external blockchain system will be treated as an actor in the text narratives below.

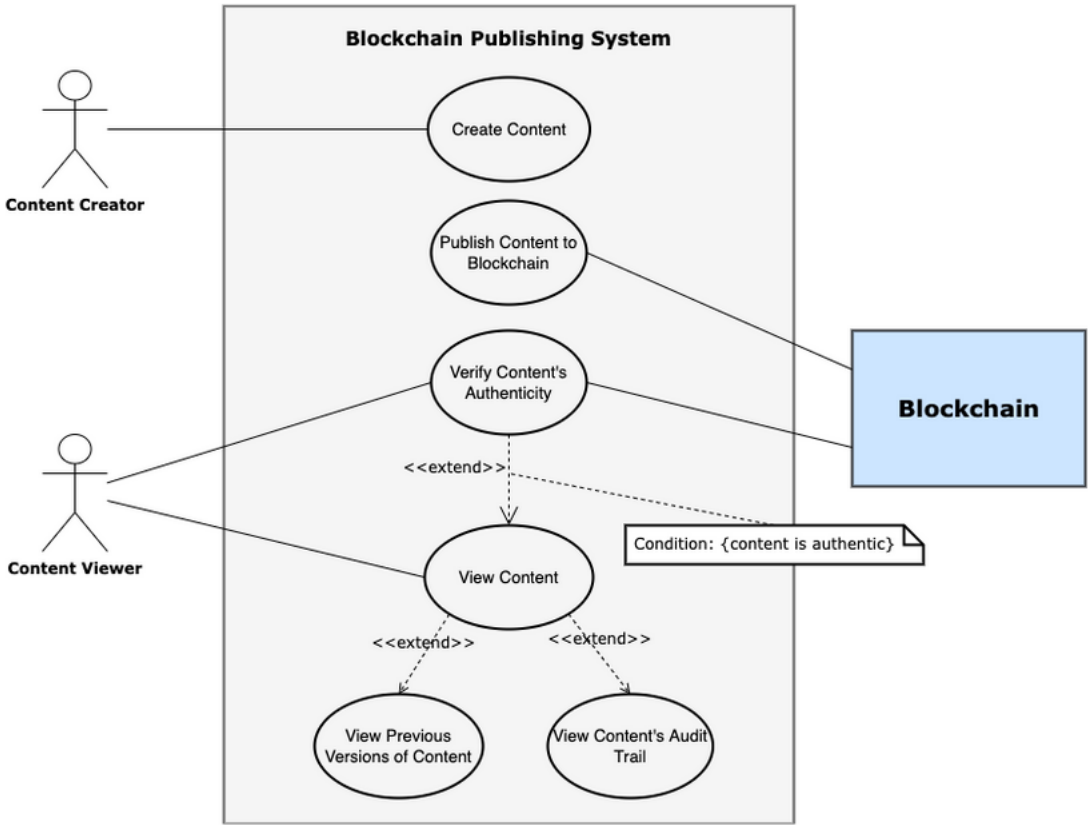

Figure 5: Use case of Create Content

Textual Description of "Create Content":

Actors involved: Content Creator

Precondition: An individual or organisation wishes to create a piece of content (e.g a blog post) to be published by the system.

Normal Scenario: The content creator creates a piece of content and submits it to the system.

Post Condition: The content is published to the blockchain.

Error Scenario: The system crashes and the content is lost and not passed into the system.

Textual Description of "Publish Content to Blockchain":

Actors involved: Blockchain

Precondition:

- The content creator has submitted some form of content to the system to be published.
- The system has access to the blockchain network.

Normal Scenario: A new block which includes the content data, metadata, and previous hash of the block these are all added to the blockchain.

Special Scenario: If there are no prexisting blocks, the block created will be a **Genesis** block, this block does not store a hash of the previous block, as there is no previous block.

Post Condition: The content is now stored on the blockchain and can be accessed by the system and published for user's to view.

Error Scenario: The new block cannot be added to the blockchain due to lack of funds or error within the system.

Textual Description of "Verify Content's Authenticity":

Actors involved: Content Viewer, Blockchain

Precondition:

- Content has been published to the blockchain.
- The block containing the content on the blockchain has not been tampered with.

Normal Scenario: The system has reference to the block on the blockchain which contains the content the user wishes to view.

Post Condition: The verified content can be viewed by the user through the web application.

Error Scenario: The block has been tampered with and the content can no longer be deemed valid or authentic.

Textual Description of "View Content":

Actors involved: Content Viewer

Precondition:

- The content viewer has selected a piece of content they wish to view.
- The content has been authenticated by the system.

Normal Scenario: The user accesses the web application and chooses a piece of content they wish to view. That content is then displayed to the user.

Post Condition: The user has access to verified content and can view previous versions of the content as well as the content's audit trail.

Error Scenario: The content is wrongly deemed valid due to a system error and the tampered content is presented to the user.

Textual Description of "View Previous Versions of Content":

Actors involved: Content Viewer

Precondition:

- System error causes previous versions of the content to be lost and inaccessible to the user.
- Each of these versions has been verified.

Normal Scenario: The user can switch between different versions of the piece of content they are viewing and see the differences between each version, as well as who made the changes.

Post Condition: The user has an overview of how the content came to be and by whom.

Error Scenario: System error causes previous versions of the content to be lost and inaccessible to the user.

Textual Description of "View Content's Audit Trail":

Actors involved: Content Viewer

Precondition:

- There is at least one version of the content published to the system.
- The changes made, as well as when the changes were made and by whom are stored in the blockchain.

Normal Scenario: The user can see a list of the various versions of the piece of content as well as the type of change made (e.g creation, edit), the date, and the author.

Post Condition: The user can access the various versions of the content through the clicking on the relevant version on the audit log as well as get an overview of how the content came to be.

Error Scenario: Changes to the content are lost or unverifiable due to system error and the information shown in the audit log is inaccurate.

3.4.3 Object Model

3.4.4 Dynamic Model

This is the dynamic model for the system.
When the user accesses the frontend, they can either request to publish or verify a document on the blockchain.
The backend updates and publishes the document to the blockchain or verifies the requested document.
The backend then returns the information about the supplied document (audit trail, past versions) or about the updated blockchain (location, owner) to the user.

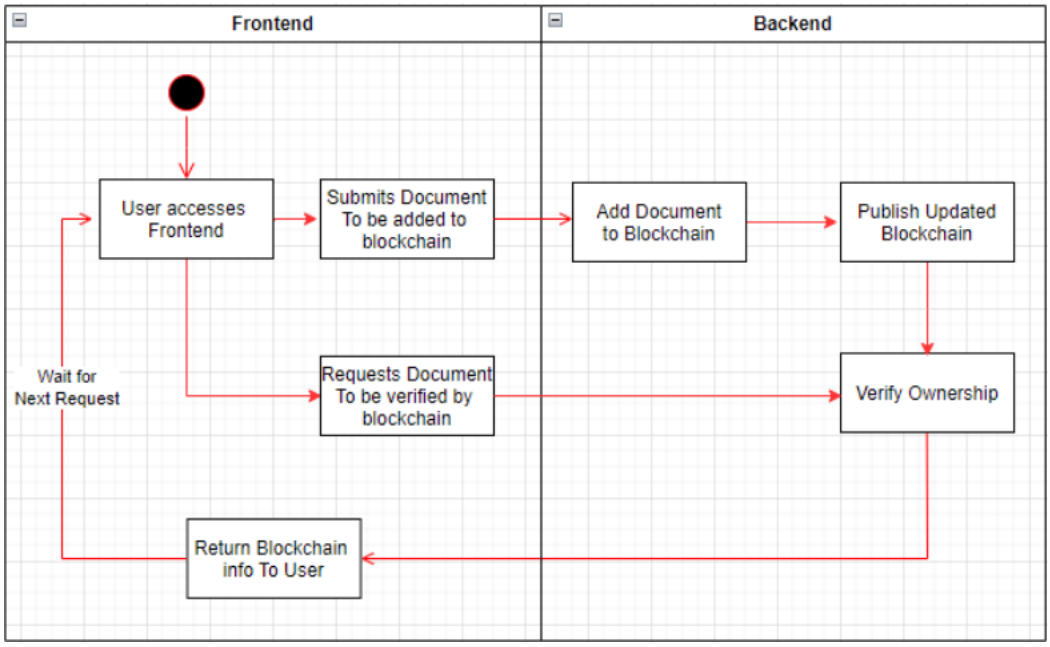

Figure 6: Dynamic model