PhiQuest Final Presentation

Allen Cheng, Ethan Wuitschick, Arnaldo Barea
Department of Computer Science and Engineering
University of South Florida
Tampa, FL 33620
allencheng@usf.edu
wuitschicke@usf.edu
ajbarea@usf.edu

1

Acknowledgments

- **We would like to thank Andre Roberts and Jay Dinicola for advising us on the project's direction and giving tips and guides on website design and building webapps in Node.js**
- **Dr. Christensen for providing feedback and tips on various documentation.**

2

Agenda

- **Background**
- **Problem**
- **Requirements**
- **Demo**
- **Design**
- **Constraints**
- **Applicable standards**
- **Trade-offs made**

Background

- **OnTask.io is a business process automation tool that allows users to easily create digital documents and forms for review and approval.**
  - Users can create workflows, a series of programmed steps to process an electronic document.
  - Competitor of DocuSign.
- **DocBridge.io is a Google Chrome extension that uses OnTask.io API calls to start a workflow with any document browsed online.**
  - Extension originally created by USF students that worked with PhiQuest and later managed and maintained by the PhiQuest team.

Problem

- **Can a private blockchain system be implemented to certify transactions and documents signed via the DocBridge.io user interface? (PS1)**

- **Can an administration user interface be implemented so that operations personnel can visualize Docbridge.io metrics, perform user administration, and log selective events and transactions? (PS2)**

Requirements

- **Problem Statement 1 Requirements**
  - As a user of DocBridge.io I want to be able to tie user signatures for a document to a private blockchain so that I may know that the signature is authentic.

  - As an administrator I want a blockchain explorer so that I can search for real-time and historical information related to the transactions that are stored in the private blockchain.

  - As an administrator, I want the blockchain to be a private blockchain that restricts who holds the ledger so that documents and signatures are kept private within the system and cannot be publicly viewed.

Requirements

- **Problem Statement 2 Requirements**
  - As an administrator I want a back-end administration interface for the DocBridge.io system so that I can edit information of administrators that are using DocBridge.io.

  - As an administrator I want a back-end visualization interface for the DocBridge.io system so that I can visualize DocBridge metrics.

  - As an administrator I want a back-end administration interface for the DocBridge.io system so that I can view and edit information of all users of DocBridge.io.

  - As an administrator I want a back-end administration interface for the DocBridge.io system so that I can view information on previous operations.

Demo

- **Demo for Problem Statement 1 Requirements**
  - Private blockchain is restricted to admin accounts that are both authenticated through Google oAuth and its account information is within the database table.
  - Blockchain Explorer displays the document identifier of documents stored in the blockchain.
  - An administrator can search a document in the blockchain by the document identifier or by a file upload.
  - Documents signed using the DocBridge.io Extension create a new transaction within the blockchain and are verifiable on the blockchain explorer. Any changes to the document will not match the document stored in the blockchain.

Demo for Problem Statement 1

Demo

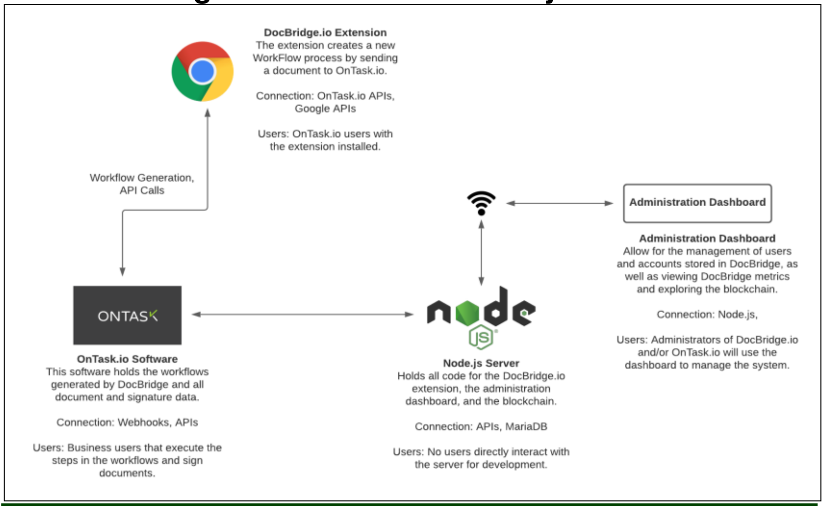
- **Demo for Problem Statement 2 Requirements**
  - Admin Management page with the ability to add, remove or edit an administrator's account information.
  - Metrics page visualizing the number of users, and how many documents have been stored within the blockchain.
  - User Management page with the ability to edit a user's account information, any groups or workflows under that corresponding user, and the ability to reset a user's API key.
  - Transaction View page showing any previous operations of users on the DocBridge.io Extension. (When a workflow starts, when groups are created/deleted, any program failures, etc.)

Demo for Problem Statement 2

9

Design

- **Block Diagram of the Overall Project**



DocBridge.io Extension
The extension creates a new WorkFlow process by sending a document to OnTask.io.

Connection: OnTask.io APIs, Google APIs

Users: OnTask.io users with the extension installed.

Administration Dashboard
Allow for the management of users and accounts stored in DocBridge, as well as viewing DocBridge metrics and exploring the blockchain.

Connection: Node.js,

Users: Administrators of DocBridge.io and/or OnTask.io will use the dashboard to manage the system.

OnTask.io Software
This software holds the workflows generated by DocBridge and all document and signature data.

Connection: Webhooks, APIs

Users: Business users that execute the steps in the workflows and sign documents.

Node.js Server
Holds all code for the DocBridge.io extension, the administration dashboard, and the blockchain.

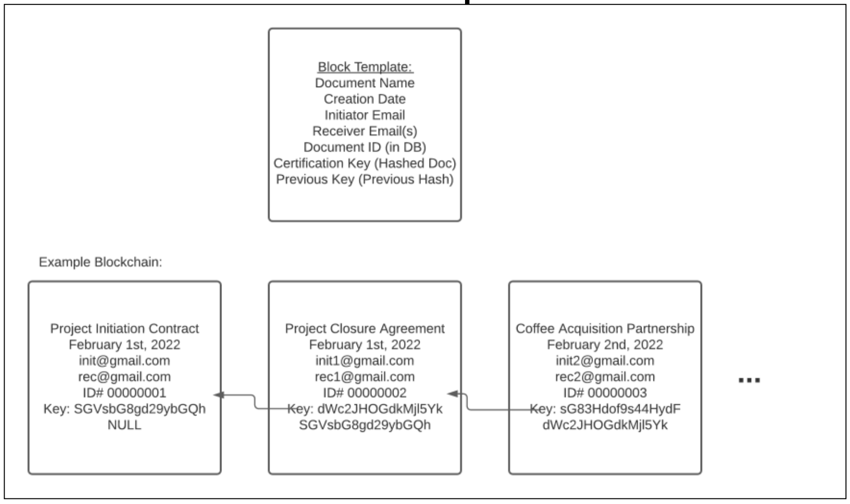
Connection: APIs, MariaDB

Users: No users directly interact with the server for development.

10

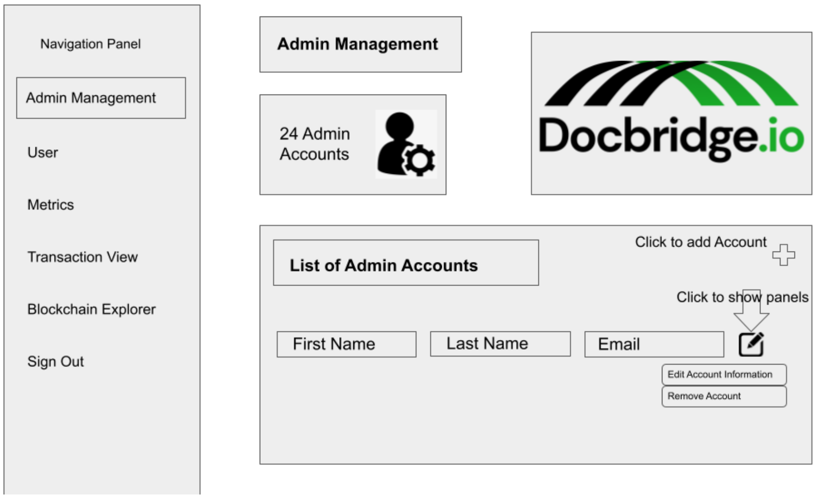
Design

- **Blockchain Structure Example**



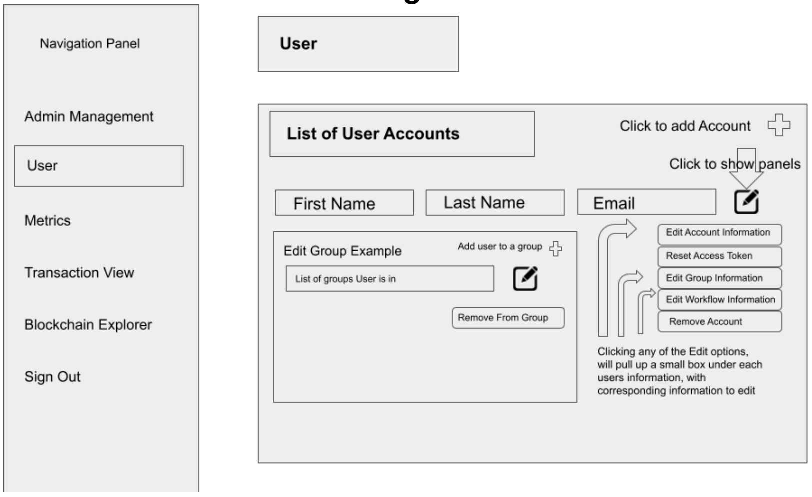
Design

- **Admin Management Page Wireframe**



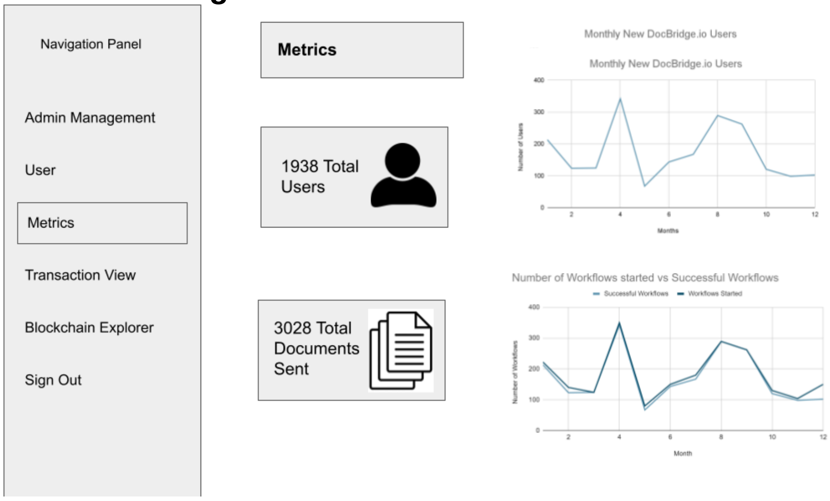
Design

- **User Administration Page Wireframe**



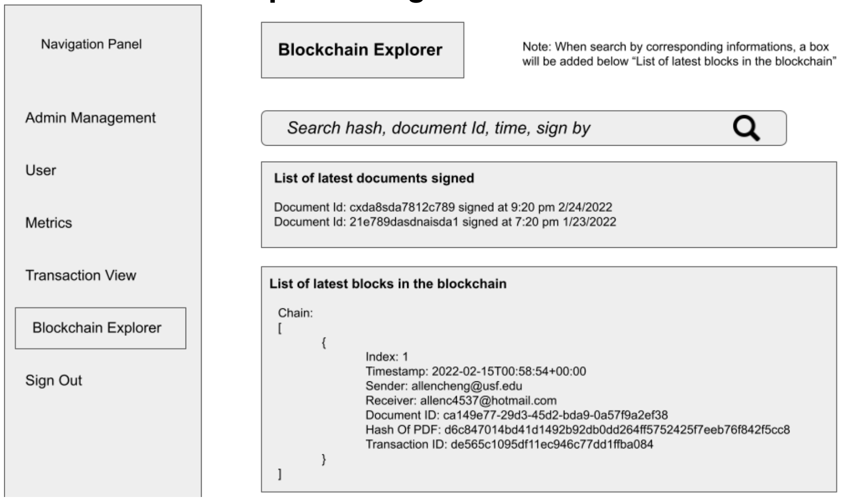
Design

- **Metrics Page Wireframe**



Design

- **Blockchain Explorer Page Wireframe**



Navigation Panel

Admin Management

User

Metrics

Transaction View

Blockchain Explorer

Sign Out

Blockchain Explorer

Note: When search by corresponding informations, a box will be added below “List of latest blocks in the blockchain”

Search hash, document Id, time, sign by

List of latest documents signed

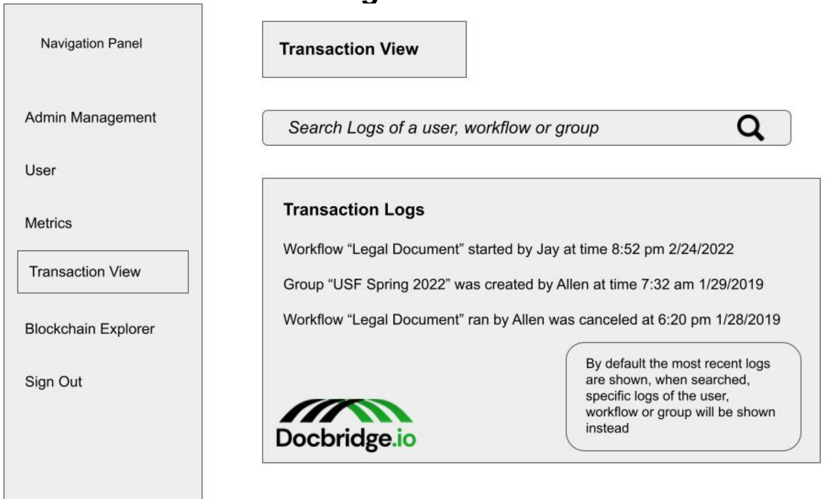
Document Id: cxda8sda7812c789 signed at 9:20 pm 2/24/2022
Document Id: 21e789dasdnaisda1 signed at 7:20 pm 1/23/2022

List of latest blocks in the blockchain

```
Chain:
[
        {
                Index: 1
                Timestamp: 2022-02-15T00:58:54+00:00
                Sender: allencheng@usf.edu
                Receiver: allenc4537@hotmail.com
                Document ID: ca149e77-29d3-45d2-bda9-0a57f9a2ef38
                Hash Of PDF: d6c847014bd41d1492b92db0dd264ff5752425f7eeb76f842f5cc8
                Transaction ID: de565c1095df11ec946c77dd1ffba084
        }
]
```

Design

- **Transaction View Page Wireframe**



Navigation Panel

Admin Management

User

Metrics

Transaction View

Blockchain Explorer

Sign Out

Transaction View

Search Logs of a user, workflow or group

Transaction Logs

Workflow “Legal Document” started by Jay at time 8:52 pm 2/24/2022

Group “USF Spring 2022” was created by Allen at time 7:32 am 1/29/2019

Workflow “Legal Document” ran by Allen was canceled at 6:20 pm 1/28/2019

By default the most recent logs are shown, when searched, specific logs of the user, workflow or group will be shown instead

Docbridge.io

Constraints

- **Document data is needed for entry into the blockchain.**
  - This data may only be accessed through the OnTask.io API, which requires the document's ID and a secure authorization token for retrieval.
- **Project must interoperate with the DocBridge.io system.**
  - The code for this system was built by previous student projects and is not well-documented for shared usage.
- **Administration of user accounts requires access to Accusoft account administration tools and privileged access to their servers**
  - Accessibility and functionality is restricted due to the smaller scope and external nature of the project.

Applicable standards

- **IEEE 1233-1998**
  - Developing System Requirements Specifications
    - Our requirements need to be understood by the project stakeholder team and by our classmates. The stakeholders have a much more intimate knowledge of the technical specifications for the project, but our class peers need to understand the different choices we have made in development so that our solutions are reasonable to that audience.
- **IEEE 730-2014**
  - Standards for Software Quality Assurance Plans
    - This project follows the agile development methodology, with client meetings occurring twice every week to QA the project software at that stage.
- **IEEE 1008-1987**
  - Standard for Software Unit Testing
    - The development of this software is test-driven, with each stage intended to satisfy some set of pre-written tests. Once these tests execute with success, the stage is considered completed. This complies with the IEEE 1008-1987 standard for software unit testing.

Trade offs

- **Reliability vs Cost**
  - The project has the blockchain that stores information about the documents signed, and a database that stores the information stored in the blockchain, which is an example of redundancy to increase the reliability of the data stored, however increasing the project cost due to server maintenance.
- **Security vs Reliability/Performance**
  - In order to verify if a document has been altered after being stored inside the blockchain, a way to generate a hash dependent on the file itself was needed.
  - Two solutions, using the metadata of the document (local download needed), using array buffer of the document (worse performance due to longer string length).

Any Questions