PhiQuest Design Review

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

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 building webapps in Node.js.**

Agenda

- **Background**
- **Problem**
- **Requirements**
- **Design**
- **Constraints**
- **Applicable standards**
- **Risk analysis and mitigation**
- **Project plan**

Background

- **OnTask.io is a web-based application for business logistics and processes.**
  - OnTask.io works with workflows, a series of programmed steps that include document signatures.
  - Workflows often start with the upload of a file, usually a document that will match a “template”.
- **DocBridge.io is a Google Chrome extension that allows for exporting a pdf into a workflow within OnTask.io.**
  - Any pdf or document browsed in Chrome, whether stored locally or found online, can be exported to start a workflow.

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

PS1 Requirements:

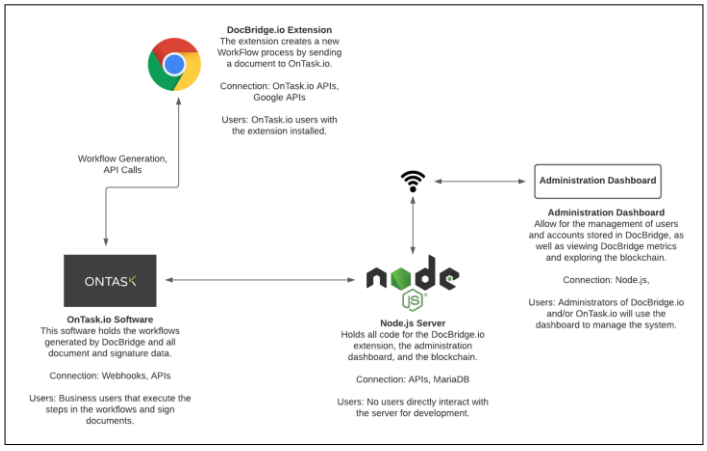
1.As a user of DocBridge.io I want to be able to tie user signatures for a document to a private blockchain so that I may know that the signature is authentic.
2.As an administrator I want a blockchain explorer so that I can search for real-time and historical information related to the transactions that are stored in the private blockchain.
3.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.

PS2 Requirements:

1.As an administrator I want a back-end administration interface for the DocBridge.io system so that I can edit information of users that are using DocBridge.io.
2.As an administrator I want a back-end visualization interface for the DocBridge.io system so that I can visualize DocBridge metrics.
3.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.
4.As an administrator I want a back-end administration interface for the DocBridge.io system so that I can view information on previous transactions.

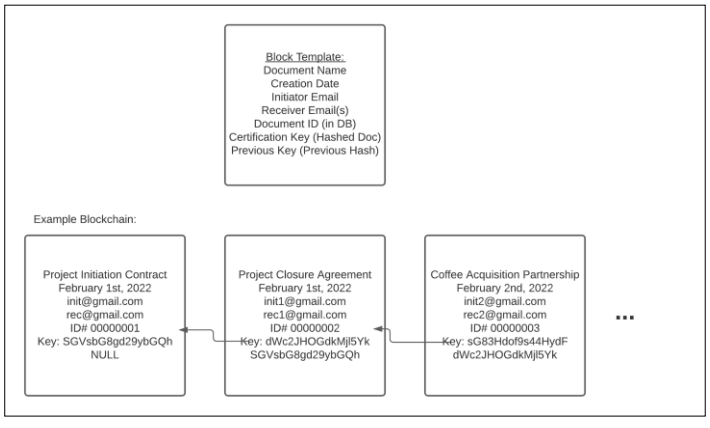
Design

- **Block Diagram**



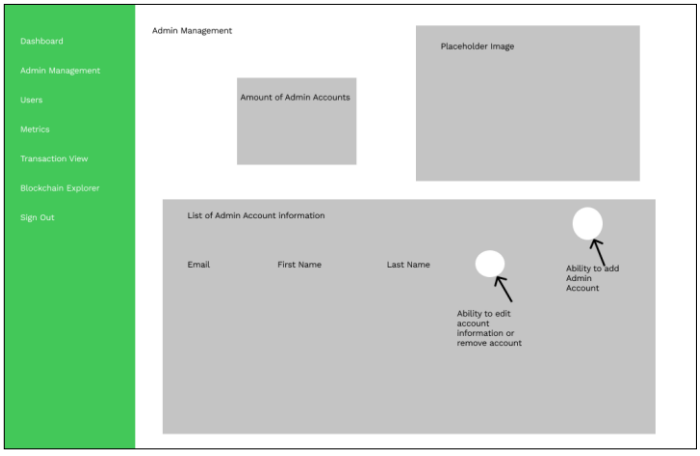
Design

- **Blockchain Structure Example**



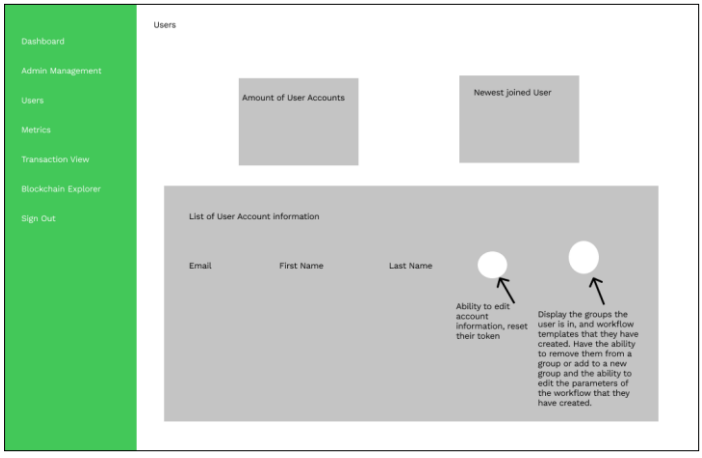
Design

- **Administration Dashboard Wireframes**
  - Admin Management Page



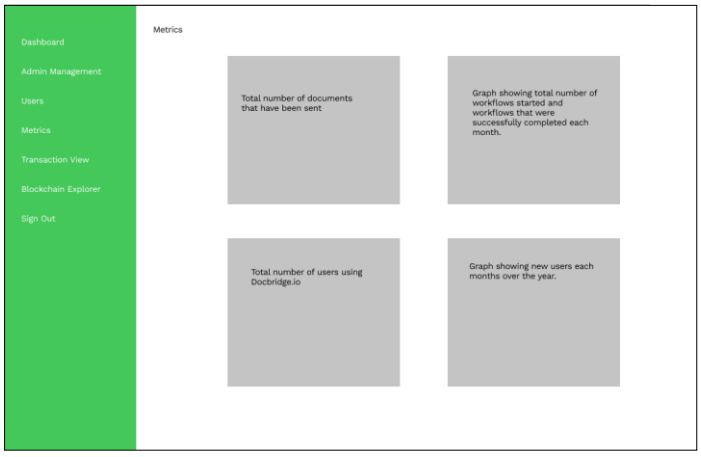
Design

- **Administration Dashboard Wireframes**
  - User Administration Page



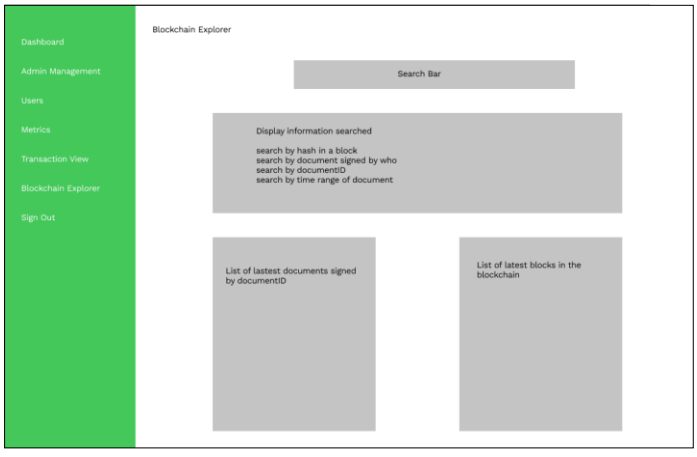
Design

- **Administration Dashboard Wireframes**
  - DocBridge.io Metrics Page



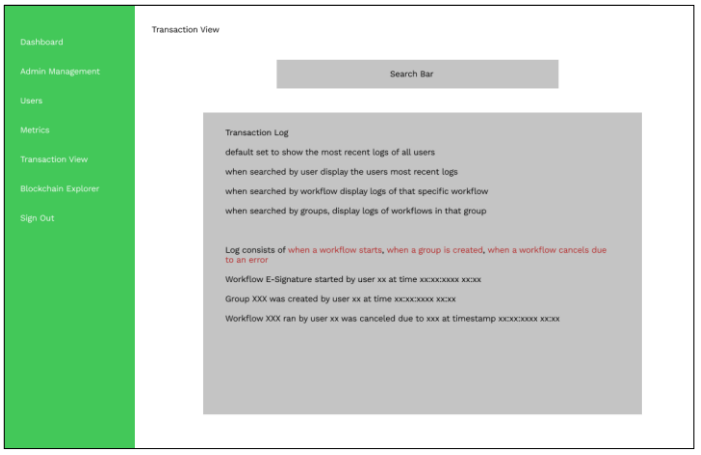
Design

- **Administration Dashboard Wireframes**
  - Blockchain Explorer Page



Design

- **Administration Dashboard Wireframes**
  - Transaction View Page



Constraints

- **Webhooks must be implemented in workflows by the OnTask.io users themselves.**
  - Users must choose to submit their documents to the blockchain, meaning that the systems will not see universal use.
- **Access to the documents in OnTask.io requires a secure access token that we cannot reliably get.**
  - This “group access token” is specific to organizations and users.
  - The token is manually created, can only be copied at creation, and can be revoked.
- **We must use existing code created in a past student project.**
  - This code is poorly commented and documented, so it is challenging for interoperability.

Applicable standards

- **Standards for Software Quality Assurance Plans**
  - The project follows an Agile development process to ensure that sufficient planning is performed for each sprint and that checkpoints are set up accordingly to evaluate the quality of the project at project stages.
- **Standard for Software Unit Testing**
  - The project uses test driven development to ensure that the end goals are meet. “Jest” is the JavaScript Testing Framework used for the project.

Risk analysis and mitigation

- **The blockchain may go unused if users don’t add the optional webhooks to their workflows.**
  - Mitigating by demonstrating the value of having hashed confirmations of documents to ensure authenticity.
- **The administration dashboard may have its functionality limited by restrictions in place by OnTask.io.**
  - Minimizing by using the OnTask.io API to expand the availability functionality.
- **We may be unable to grab the necessary secure tokens to access documents, breaking the functionality of the blockchain.**
  - Reassured by PhiQuest that it will not be an issue in the future.

Project plan

Task	1/28	2/4	2/11	2/18	2/25	3/4	3/11	3/18	3/25	4/1
Setup Website										
Research Blockchain										
Implement Blockchain										
Implement Certification										
Implement Admin Acct										
Implement User Profile										
Implement BC Explorer										
Implement Metrics										