

AggieSTEM Library Final Report

Team NewBee

May 2020

Project Summary	2
Team Roles	2
User Stories	2
Dataset Query (Iteration 1)	2
Online File Preview (Iteration 2)	2
Server File Storage (Iteration 3)	3
Online Request Process (Iteration 4)	3
Data Request Form Update (Iteration 4)	3
Test Coverage (Iteration 1-4)	3
Iteration Progress History	3
Iteration 0	3
Iteration 1	3
Iteration 2	4
Iteration 3	4
Iteration 4	4
Lo-Fi UI	5
Screen Captures of New Features	6
File Search	6
Request History	7
Request Process Portal	7
File Online Preview	8
Legacy Project Summary	8

Project Summary

AggieSTEM Library is a website that manages user access to dataset files. Users may submit a dataset request to the website, and the administrator decides whether the request is accepted or declined. With approval, users may gain access to a dataset file hosted on the server. A search function is provided to the user to efficiently find the target file. They may preview dataset files online, or download to their local disk. On requests for dataset files not available on the server, administrators may upload the files through a portal, which are kept track of by the server database. Users may also follow the progress by checking out a history page that collects all their past requests and status updates.

Team Roles

- Scrum Master: Luming Xu
- Product Owner: Gehao Yu

Team Member Name	Email
Luming Xu	xuluming@tamu.edu
Yuchen Jiang	yuchen.jiang@tamu.edu
Yuhan Cheng	chengyh971030@tamu.edu
Gehao Yu	gary19961231@tamu.edu
Yu Shen	shenyu.tcv@tamu.edu
Xinpu Liu	liuxinpu16@tamu.edu

User Stories

1. Dataset Query (Iteration 1)

As a user granted access with multiple data files,
So that I can find the target data efficiently,
I want to **search** for data by its name.

2. Online File Preview (Iteration 2)

As a user with access to a certain data file,

So that I can see if the data is what I am looking for,
I want to **preview** the data on the website. (table, pdf, image, text support)

3. Server File Storage (Iteration 3)

As an administrator of the website,
So that I can assign dataset files to users that filed in a request,
I want to **upload** files to the server.

4. Online Request Process (Iteration 4)

As an administrator of the website,
So that I can **process** user requests for data access,
I want to make accept/decline decisions online based on data request forms.

5. Data Request Form Update (Iteration 4)

As a user who has possibly filed in **multiple** user requests,
So that I can keep track of all the records,
I want to easily check for status update on each request

6. Test Coverage (Iteration 1-4)

As a test engineer,
So that I could test new versions in the future,
I want to **read** a test case of existing functions to see if any bugs have happened.

Iteration Progress History

Iteration 0

We met with the customers, Professor Capraro, along with the previous developer Donald Beyette, to get down to the project and make clear of the features that are most needed to the website. We also drew Lo-Fi user interfaces, and sketched several user stories.

Meeting date: Feb. 24, 2020, at Prof. Capraro's office.

Iteration 1

We wrote several tests to familiarize with the legacy project, also to set up a baseline of the existing features. A dataset query feature is implemented.

Iteration 2

An online file preview feature is added that supports multiple file formats. We demonstrated the currently available features to the customers, and they gave quite positive feedback.

Meeting date: March 24, 2020, via Zoom

Iteration 3

A upload portal and file database are set up on the server, which prepares for the online process of data requests. Several tests are added to make sure of correct feature behavior.

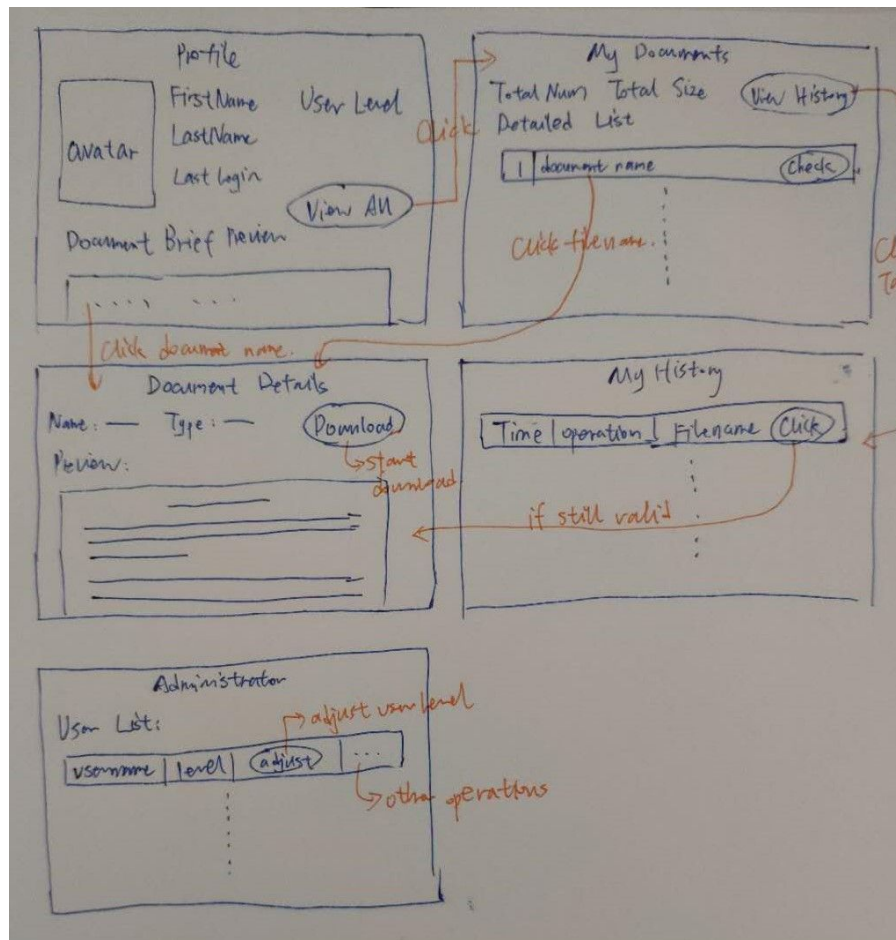
Meeting date: April, 4, 2020, via Zoom

Iteration 4

An online request process feature is fully implemented. Administrators now can handle data requests online, and assign dataset files access to users.

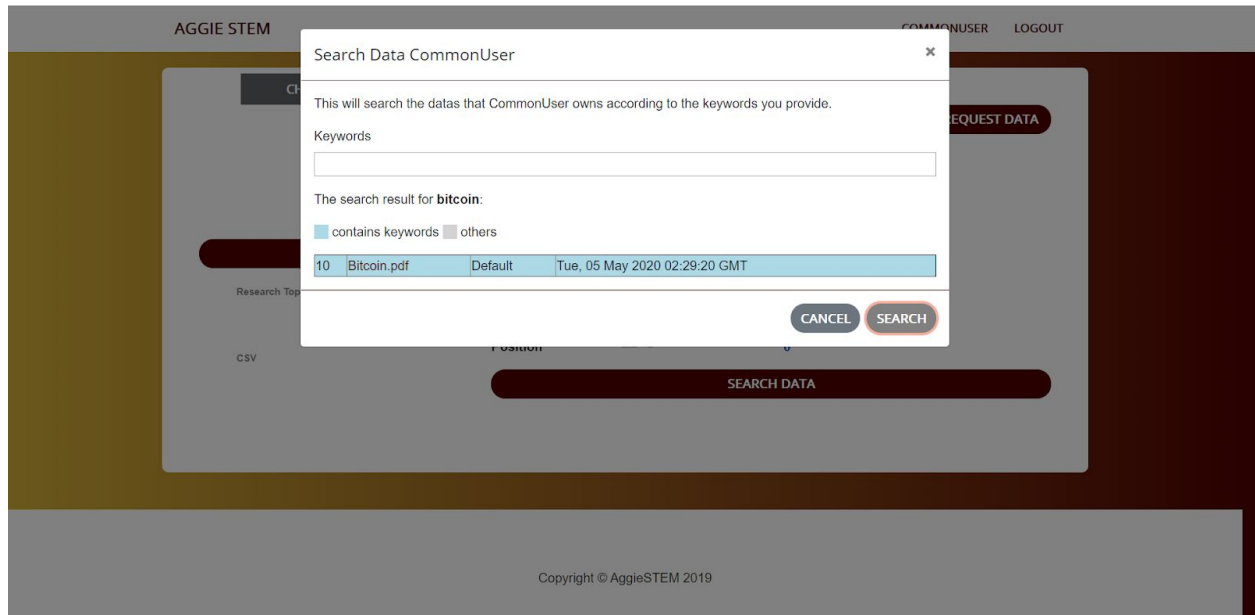
Meeting date: April 23, 2020, via Zoom

Lo-Fi UI



Screen Captures of New Features

File Search



AGGIE STEM

COMMONUSER

LOGOUT

Request History

New Request

Status: Approved		Delete
Create Time	Tue, 05 May 2020 02:23:57 GMT	
Data Elements	Bitcoin: A peer to peer electronic cash system	
Requestor Info		
First Name	Common	
Last Name	User	
Organization / Institution	ECEN	
Status	Student	
Phone Number	9794221234	
Email Address	commonuser@tamu.edu	
Address	College Station	
Data Usage		
Research Topic	N/A	
Co-researchers / authors	N/A	
How the data will be used	N/A	
Data Access Details		
Access Name	N/A	
Access Email	N/A	
Access Institution	N/A	
Access Phone	N/A	

AGGIE STEM

demo.csv		Approve
reason		Decline
Create Time	Tue, 28 Apr 2020 14:33:52 GMT	
Data Elements	MNIST	
View Detail		

demo.csv		Approve
reason		Decline
Create Time	Tue, 28 Apr 2020 15:24:30 GMT	
Data Elements	IMDB database	
View Detail		

demo.csv		Approve
reason		Decline
Create Time	Tue, 28 Apr 2020 16:00:37 GMT	
Data Elements	imdb review	
View Detail		

demo.csv		Approve
reason		Decline
Create Time	Tue, 05 May 2020 02:05:09 GMT	
Data Elements	I request a MNIST dataset.	
View Detail		

MANAGE USERS
MESSAGE USERS
ADMIN
LOGOUT

demo.csv		Approve
reason		Decline
Create Time	Tue, 28 Apr 2020 15:24:18 GMT	
Data Elements	MS-COCO	
View Detail		

demo.csv		Approve
reason		Decline
Create Time	Tue, 28 Apr 2020 15:24:41 GMT	
Data Elements	demo.csv demo1.jpg demo2.jpg demo3.jpg demo.pdf demo.txt imdb.jpg May_2_2020.txt imdb.jpg Bitcoin.pdf demo.txt	
		Approve
		Decline

demo.csv		Approve
reason		Decline
Create Time	Tue, 05 May 2020 02:23:57 GMT	
Data Elements	Bitcoin: A peer to peer electronic cash system	
View Detail		

File Online Preview

Aggie STEM

COMMONUSER LOGOUT

Current Dataset: Bitcoin.pdf

Page: [1-9] 5d

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshi@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and overpower attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a transfer cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, handing them for more information than they would otherwise need. A certain percentage of funds is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from double-spending attacks and escrow arrangements could be implemented to protect buyers. In

Next

Legacy Project Summary

We have received timely support from the legacy project developer Donald Beyette during the development within this semester. Much thanks to DJ for his effort.

The legacy project has implemented user sign up and log in features. Two user groups, admin and common user are set up, and the website contains reactive front-end styles. A data request form submission page is also provided for the common users.

Link to legacy code repo:

<https://github.com/unkown512/AggieSTEM-DL-PRODUCTION/tree/42c2746256aeaae348648935b144cf967d425a92>