Michael Shull
631 North Stephanie St.
Henderson, NV 89014
mshull@g.harvard.edu
(541) 864-9285

Proposed Start Date
August 30, 2016

Anticipated Date of Graduation
March 2017

Faculty Approval
1. Tentative Project Title

Address Infinite is the tentative title for my Capstone project. Below is a rough mock-up of the application logo.

![Address Infinite Logo](image)

2. Abstract

Whether it’s a desktop computer, laptop or tablet, a single person owning multiple devices and/or computers is becoming very common. To work with multiple platforms some of today’s software and services may still require specific hardware or the purchase of multiple subscriptions. Address Infinite attempts to address these problems for one of the most common applications used today, the digital address book.

Capable of running on any modern Mac, PC or Linux desktop computer with an Internet connection, the goal of Address Infinite is to provide a well organized and accessible address book that requires no paid subscription or single operating system. To accomplish these goals Address Infinite will be created using the Java programming language, a cross-platform language, as well as leveraging a new technology called the “blockchain”. Address Infinite will provide several useful features which include a contact list manager, note-taking, scheduling, historical logging and integration with third-party communication and mapping tools.

Address Infinite will leverage blockchain technology to provide free data on a distributed platform. Address book information is never lost or inaccessible due to server outages and anyone can backup their data without need of a credit card to pay for a third party service.

The final Capstone deliverables will include:

- Mac, PC and Linux executable files which run the Address Infinite desktop application.
- A user manual in PDF format which provides instructions and application details.
- An archive of the application source code and final Photoshop and Illustrator design files.
3. Project Description

3.1 Project Overview

Address Infinite will provide address and scheduling tools similar to those provided by Apple Contacts and Windows Address Book. The application will be compatible with Apple, PC and Linux desktop computers.

Some key differences between the Apple and Windows address book software and the Address Infinite software include:

- A modern set of visual interfaces that leverage grid layout content placement and flat design techniques.

- Multi-platform compatibility while maintaining a consistent user experience regardless of the operating system.

- The ability to access your address book information on any desktop computer with an Internet connection without the need for a subscription to a third-party service or organization.

- Unique features typically found in paid address book products such as multi-user access and cross-platform syncing.

This application leverages a group of technologies collectively called the blockchain. The purpose of the blockchain technology, as it relates to Address Infinite, is that it provides free data storage across a distributed platform, while also eliminating the problem of having a single point of failure. One’s data is never lost or inaccessible due to server outages, and anyone can backup their data without need of a credit card to pay for a third party service.

3.2 Application User Interfaces and User Experience

Address Infinite will utilize grid layout and flat design techniques to create a graphical user interface (GUI) for the application. This application will be composed of six sections, below is a breakdown of each individual section along with mock-up design compositions to provide a rough visual example of the proposed layout. (Cousins C., 2013, *The Ultimate guide to flat design.*)
All Address Infinite interfaces will be designed using a combination of vector and rasterized artwork created using Adobe Photoshop and Adobe Illustrator.

- **Login & Registration Interface**

  The login interface provides a 6-digit pin password field for the user to enter to gain access to the application. For new users, a registration button is provided which takes the user to a registration form where they can enter their data (used as the first address book entry) along with setting their 6-digit security pin.

  ![Login Interface Diagram]

  Technical Note: The blockchain has its own built-in security so this PIN login system is only for the convenience of the Address Infinite GUI application. Anyone running a custom version of the Address Infinite protocol does not need a PIN to access Address Infinite data if they already know the address and cryptographically secure private key for the data they are looking to access.

- **Universal Design Elements**

  Once a user enters the address book there will be a few common design elements that expand across all of the interfaces within the application (barring the login and registration). These design elements include navigation tabs and a search box at the top of the application. Certain data related to the blockchain network will be displayed at the bottom of the
application to allow the user to easily see the status and health of the Address Infinite blockchain network.

• **Dashboard**

The address book dashboard is the first interface the user experiences after logging into the application. This interface provides an organized snapshot of the data within the address book. The dashboard prioritizes data that is urgent, recently updated, recently added or frequently accessed. The goal of the dashboard is to give the user quick access to contacts, schedules or notes that the user is likely to need most often. The design of the address book dashboard will be broken up into 5 fluid sections.
• Contact List

The contact list interface provides a searchable index of contacts that can be viewed individually. The list will be within a scrollable table, allowing the user to easily find the contact they are looking for. A limited number of filters will be available to help reduce the list of potential contacts. These filters include such things as contact tags, organizations and alphabetical sorting.

In addition to the contact information, each contact page has the ability to manage contact notes. These notes can be appended to bottom of the contact information in descending order.
• **Schedule Interface**

Address Infinite provides a simple calendar and event entry system to keep a schedule that is in sync with your contact list. The schedule interface helps users keep track of important events and assists in future planning.
• **Logging Interface**

The historical logging interface provides a linear history of logs regarding high-level actions that have taken place within the application. This information may provide useful when needing to roll-back a particular change to a contact, note or scheduled event.

![Log Filter & Tools](image)

- **Blockchain Token Generator Controls**

These controls provide a simple on or off switch for generating Address Infinite blockchain tokens. These tokens allow computers to pass address book data back and forth to each other securely. When the generator is turned on, your computer will run software to generate “tokens” that are then sent to other machines running Address Infinite. Attached to these tokens are unreadable encrypted pieces of your address book data. Upon your request, these other machines will send tokens back to you with your address book data in the same way Dropbox or Google Drive might send you your files. You may choose to generate as many tokens as needed or turn off the generator to save processing power. (Satori M., DeRidder C., Grosser J., May 12, 2016, *Blockchain Basics: A Primer*)

- **Blockchain Account Interface**

The account interface is the blockchain dashboard of the Address Infinite application. This
interface provides a snapshot view of the various blockchain accounts you have in your possession, the number of tokens each account possess and the amount of address book data you have saved into the blockchain. Tokens can be transferred to other accounts from this interface and navigation is provided to access other Address Infinite tools such as the Blockchain console.

- **Blockchain Console**

  The blockchain console is a technical interface for blockchain-specific interaction which allows the user to view raw blockchain logs to see exactly what is going on in the back-end of the application. The blockchain console also provides a command-line interface for simple blockchain queries such as viewing another’s token count or viewing the number of records they may have (the data itself is encrypted).
3.3 Application Objective

The end objective of this application is to provide a well organized and designed address book that can run on any major operating system and requires no paid central authority network (Google, Amazon, etc) to sync online.

In addition to the Address Book, this application will be used to demonstrate a real-world product for blockchain technology that is outside of the financial services space.

3.4 Blockchain, More In-Depth

Paradigm shifts are at the core of modern computer technology. Ranging from the first electromechanical computer during World War Two to the recent Internet revolution, every step of computing has required changes in the way we think about hardware and software. Today, the Internet is a ubiquitous service spanning the entire world, reaching billions of people with billions more still to go. The Internet is an expected underpinning of modern computing and due to this demand we are beginning to see the emergence of next-generation software, software that builds upon a constantly connected, ever-improving, Internet. One of the most exciting new software paradigms that has emerged from this ever-connected environment is the “blockchain”.

The blockchain is a general term used to describe a collective set of technologies that, when combined together, create a software platform and protocol for a decentralized and immutable database. A database that can run on any machine with an Internet connection and adequate
hardware. This simple database technology has an unlimited number of possible applications due to its unique decentralized and immutable properties. In other words, this database software lives on many computers and cannot be changed by one computer without the majority approval of the other computers. The “protocol” portion of this software provides a strict set of rules that must be adhered to if you want to change the database. If you follow the rules, the other computers will likely accept your change and update their version of the database with your changes. This process allows data to be saved across many computers while preventing one malicious computer from ruining the information.

The most popular software using blockchain technology at the moment is called Bitcoin, a form of digital money that can be exchanged for services and goods by those who choose to accept Bitcoin tokens, otherwise known as BTC. Although Bitcoin was one of the first blockchain-based technologies, the potential for blockchain far exceeds just the financial space. Address Infinite is designed to leverage the networking and database consensus functionality of the blockchain to create one of the few non-financial applications that use these features. (Nakamoto S., 2008, *Bitcoin: A Peer-to-Peer Electronic Cash System.*)
3.4.1 Security of the Address Book

Address Infinite uses U.S. government adopted cryptographically secure algorithms to keep all address book data safe, both on your local computer and on the many other computers storing pieces of your encrypted data via the blockchain.

3.4.2 Blockchain Libraries for This Project

Although certain blockchain technology is available to the public via open source, the code provided for these projects are often highly optimized and difficult to separate from their core purpose which is typically the creation of cryptocurrency tokens. Address Infinite uses a new blockchain codebase modeled after existing methodology. This codebase will include a simple set of Java libraries that perform the various functional tasks required to make a blockchain. These libraries do not focus on optimization and instead focus on organization and reusability for maintaining a modular codebase and to assist those wishing to understand blockchain technology better.

It is important to note that Address Infinite does not re-invent blockchain methodology. Instead, it’s libraries are based off of the tried-and-true blockchain algorithms that already exist in the wild. Thankfully, most blockchain applications provide open source white papers containing the conceptual and mathematical formulas of their blockchain algorithms allowing any who wish, the ability to create custom software that use their protocols. Address Infinite adopts the existing algorithms that best match the application goals. This requires a combination of common blockchain algorithms along with some custom classes tailored specifically to the Address Infinite platform. Certain systems, such as the blockchain Ethereum implement useful and novel ways to trim the size of the blockchain while increasing the speed of transactions. (Buterin V., 2014, Ethereum White Paper.)

The Address Infinite Java library classes that will be created include:

- **Local Utility Class**

  The local utility class which handles functionality native to the computer running the software. This can include caching certain data locally, handling application login security and any general work needed that involves simple programming tasks or interacting with the local machine.
- **Cryptography Class**

Cryptography is at the heart of the blockchain. This class is used exclusively to handle hashing, encryption and decryption functionality. Due to the complex nature of this functionality, it may be necessary to include third-party libraries that have been hardened over the years through heavy use and testing. This may include such libraries as OpenSSL. (Wood G., 2014, *ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER*)

- **PeerToPeer Networking Class**

Address Infinite requires multiple “nodes” to run the Address Infinite software to operate at peak efficiency. The more nodes the network has running, the more secure the network becomes against malicious attempts to modify the blockchain data. The PeerToPeer (P2P) class provides the functionality that allows for discovery and interaction with other machines running the Address Infinite software. This functionality is consistently used to keep every node apprised of the network and status of the state of the blockchain database.

- **Consensus Class**

The consensus class works with the actual blockchain data and maintains the security of the blockchain by processing incoming data and verifying the data’s validity. This class contains the bulk of the protocol algorithms and is heavily customized to the blockchain being built, or the Address Infinite blockchain in this scenario.

- **Proof of Work Class**

The Proof of Work (PoW) is the method in which computers on the blockchain compete amongst one another to generate cryptographically secure tokens. Tokens which are used as a method of exchange across all of the computers on the network to determine which computer has the ability to update the database across the network at that moment. The more you compete, the more tokens you generate, and the more tokens you generate, the more you can spend in order to save data to the blockchain database. This class provides the methods needed to perform this token-based competition in order to keep the blockchain secure while having a mechanism of scarcity in the network which prevents malicious computers from attempting to save data to the blockchain without participating in the blockchain process.
• **Address Infinite Class**

The Address Infinite class is the control center of the application. This class contains the high-level calls that utilize the other libraries. This class will be used by the main controller files as a simplified programatic interface to interact with all of the Address Infinite back-end components.

3.5 **Course Work**

Before this Capstone I completed the following courses:

• CSCI E-52 : 2015 Fall : Intensive Introduction to Computer Science
• CSCI E-12 : 2016 Spring : Fundamentals of Website Development
• CSCI E-15 : 2016 Spring : Dynamic Web Applications
• CSCI E-3 : 2016 Spring : Introduction to Web Programming Using JavaScript
• CSCI E-90 : 2015 Fall : Cloud Computing
• CSCI S-78 : 2016 Summer : Wearable Technologies and the Internet of Things
• CSCI E-32 : 2016 Fall : Advanced JavaScript with Node.js and AngularJS
4. Technology Used for Address Infinite

Java

I propose to use the Java programming language for this project. I decided to use Java due to its popularity and virtual machine availability on every platform. Java is one of the most highly-used languages in the Enterprise space and also one of the most heavily supported programming languages in existence. Writing the Address Infinite protocol initially in Java gives the project the best odds of success in regards to mass adoption. For the interface framework I will likely use Java Swing or a comparable graphics framework to create the GUI.

Adobe Photoshop & Illustrator

The GUI portion of this application will be designed using Adobe Photoshop and Illustrator. Photoshop will be used for all non-vector artwork. All windows, buttons, and widgets will likely be custom designed within this software.

Miscellaneous Digital Artwork Applications

I will utilize a series of various small artwork applications that help manage my digital art tablet, desktop rulers and color swatches.

UDP & TC/IP Networking

This application uses peer-to-peer technology significantly. The User Datagram Protocol provides the fastest method of data transfer and is used for node discovery within the application. The Transport Protocol / Internet Protocol is used as a stable method of node to node interaction. Unlike UDP, TCP/IP verifies the data packets are received which is why this protocol is used during the actual communication process.

Cryptography

Cryptographic functions are used heavily throughout the Address Infinite application. The blockchain itself uses a customized set of code that creates a periodically pruned Merkle tree. For hashing I use Bitcoin’s very own SHA-256 double hashing method. Encryption and decryption are based on generated public and private key pairs using an Elliptical Curve algorithm.
5. How to evaluate the work?

This project can be evaluated as a working prototype application for the PC and Mac. An executable will be provided that can run on a modern desktop computer. The project should look and function as described in this document although due to the nature of blockchain technology it should not be expected to be as secure as certain applications like Bitcoin due to the limited number of Address Infinite nodes that will be running.

5.1 User Experience and Design

I suggest evaluating the design of the application first.

- Did you find the design of the application easy to understand?
- Were you able to navigate to the different areas of the application?
- Was the design efficiently organized?

5.2 Functionality

When evaluating the functionality of the application, here are a few things you may want to consider.

- Did the application work as expected?
- Was the wait time within reason?
- Were you able to use the blockchain in a way you found useful?

5.3 Notes

An aggressive owner of many Address Infinite nodes could perform what is known as the 51% attack on Address Infinite and likely cause the network to crash. Typically, production blockchain applications will launch their product with many node partners before making their system public. This gives the network adequate time to harden against node consolidation and makes the chances of attack far less likely as the attacker would need to possess the hashing power equivalent of a major corporation such as Google. For purposes of this Capstone I will only run about 2-5 dedicated nodes to adequately demonstrate the application.
6. Schedule

The table below reflects my proposed schedule of completion for my Address Infinite Capstone project.

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Estimate</th>
<th>Completion Date</th>
<th>Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capstone Proposal Approval</td>
<td></td>
<td>August 16, 2016</td>
<td>Final Draft of Capstone Proposal</td>
</tr>
<tr>
<td>Research &amp; Preliminary Design</td>
<td>1 week</td>
<td>August 17, 2016</td>
<td>Sketches of Initial Design Ideas</td>
</tr>
<tr>
<td>Preliminary GUI Design</td>
<td>1-2 weeks</td>
<td>August 22, 2016</td>
<td>Images of the 5 major application interfaces</td>
</tr>
<tr>
<td>Coding of Base Classes &amp; GUI Attachment</td>
<td>1-2 weeks</td>
<td>September 5, 2016</td>
<td>Document mapping out the structure of the application code and a wireframe describing the flow of the application.</td>
</tr>
<tr>
<td>Initial Coding of P2P and Blockchain Classes</td>
<td>3 weeks</td>
<td>September 26, 2016</td>
<td>Executable file for Mac or PC that can demonstrate the shell of the application and the ability to click through to the various interfaces.</td>
</tr>
<tr>
<td>GUI Completion with Test Data</td>
<td>2 weeks</td>
<td>October 9, 2016</td>
<td>Updated Executable file for Mac or PC that had test data within the application and more internal application widgets (scrolling, tables, lists, etc).</td>
</tr>
<tr>
<td>Completion of All Classes</td>
<td>3 weeks</td>
<td>October 30, 2016</td>
<td>Updated Executable file that appears to be fully working, only it’s running off of test data.</td>
</tr>
<tr>
<td>Internal Testing and Debugging</td>
<td>1 week</td>
<td>November 4, 2016</td>
<td>Document containing a list of all the bugs found during internal testing and the work being undertaken to correct the issues. Latest Executable file for Mac or PC also included.</td>
</tr>
<tr>
<td>Large Scale Multi-Machine Test</td>
<td>1 week</td>
<td>November 11, 2016</td>
<td>Working executable file for Mac or PC with the bugs corrected. Test data is removed and the application will be running live at this point. This will be considered the “Alpha” version of the software. A list of upcoming tests will be included.</td>
</tr>
<tr>
<td>Milestone</td>
<td>Estimate</td>
<td>Completion Date</td>
<td>Deliverables</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>Documentation &amp; Final Tweaks</td>
<td>1 week</td>
<td>November 18, 2016</td>
<td>The latest Executable file considered to be the “Beta” version of the Address Infinite software. Document containing the results of the previous tests that were undertaken will be included.</td>
</tr>
<tr>
<td>Capstone Submission</td>
<td></td>
<td>November 23, 2016</td>
<td>The final “Gold Master” version of the Address Infinite software that can run on a Mac or PC. The Address Infinite user manual in PDF format will be included. A zip file of the Java source code, PSDs and AI files will also be included.</td>
</tr>
</tbody>
</table>
7. Future Considerations

File Storage

Although not part of the Capstone project, future enhancements of Address Infinite may include the use of decentralized file storage to allow for actual binary file uploads and blockchain storage. File storage could open up many possibilities for Address Infinite and allow for less data entry and more drag-and-drop functionality.

Communication

Address Infinite could leverage the blockchain storage system to create a separate chain that contains only messaging to one another. The messages could dissipate over time providing almost a SnapChat type of functionality.

Decentralized Web

If file storage were added to Address Infinite it might be possible to create a web-like feature of Address Infinite that allows HTML, CSS and images to be hosted within Address Infinite with an internal browser for viewing such files and a protocol for navigating to these files. IPFS and MaidSafe are two similar technologies in the works today trying to accomplish this very thing.

Gaming

Decentralized gaming could be an interesting feature that might be added to Address Infinite in the future. The blockchain provides a number of mechanisms that could allow for random number generation which could lead to the creation of a turn-based game such as a virtual role playing game.

Asset Exchange

If Address Infinite gained enough popularity, the Address Infinite tokens might be in demand and would therefor represent a financial value on an open exchange. In theory you could use these tokens for extremely different assets, for example on behalf of farmers around the world. People could buy the tokens, distribute them to those in need and local farmers could redeem the tokens for food. The farmers would then put their tokens on an open exchange for other people to buy and the cycle would start over again.
8. Glossary

Bitcoin

A software application created to demonstrate a thought-experiment regarding trust-less methods of transferring digital money. The demonstration was so successful that it is considered a legitimate asset today with a current value of hundreds of US dollars per single Bitcoin or BTC. (Nakamoto S., 2008, Bitcoin: A Peer-to-Peer Electronic Cash System.)

Blockchain

A plethora of technologies that perform various tasks which enable the creation of a decentralized immutable ledger that can be securely modified and validated. The actual data that contains this ledger is called the “blockchain” due to it’s characteristic of chaining previous “blocks” of data to the latest block of data.

Consensus

When one or more parties come to an agreement or understanding. In the context of the blockchain, consensus is usually referring to the agreement between nodes regarding the state and values of the blockchain that each node has.

Cryptography

A method of securely storing or comparing information using complex mathematical algorithms.

Decentralization

The lack of a central authority or central point of failure.

Flat Design

A design methodology that promotes symmetry, simple color palettes, and a healthy use of white space in order to achieve a functional design esthetic. Good examples of successful flat design initiatives include the latest version of Google Android’s “Material Design” initiative, the Windows Metro design, and Apple’s latest iOS UIKit.

Grid Layout
Grid layout is a method of organizing design elements into grid-like patterns in order to achieve consistent spacing and fluid layouts. The typical grid layout process requires organizing data into grid units and maintaining information containment within their pre-designated grid areas.

**Java**

A server-side programming language which compiles into a virtual machine, allowing applications to run in virtually any environment.

**Peer to Peer Networking**

A decentralized method of networking where machines, otherwise known as “nodes”, connect to each other and share the communication addresses of other nodes so they may connect as well. Once connected, a dialogue can occur and possibly a shared consensus can be achieved. (Hamid N., *Peer-to-Peer Programming*)
9. References


(April 14, 2014) - MaidSafe.net announces project SAFE to the community. Retrieved from https://github.com/maidsafe/Whitepapers/blob/master/Project-Safe.md


