CENG 499

Progress Report #2

air.auth

By:
Group Number: 6
Anubhav Mishra (anubhav@uvic.ca) V00740087
Cole Bosmann (cboss24@uvic.ca) V00722585
Conrad Foucher (conrad@foucher.ca) V00721922

Supervisor: Dr. Kin F. Li
Submitted: June 14th 2014

Introduction

Our team plans to create a web-based password management service that incorporates Leap Motion technology. Like all password managers, the Air.Auth system will allow users to store their various passwords and access them using a single master authentication. However, what sets Air.Auth apart from traditional password managers is the method it uses for this master authentication. Instead of the regular username and password combination, Air.Auth will allow users to gain access to all their stored passwords by scanning their hand with a Leap Motion. Air.Auth will also allow users to record hand gestures and assign them to specific websites for which they have stored passwords. Therefore, signing into a website with stored credentials becomes a two step process, first: the user presents their hand for authentication, second: the user recreates their previously saved gesture.

In order to allow users access to their stored passwords from any computer, Air.Auth has been designed as a web service. As a result, it follows the traditional client-server architecture. The client side (nicknamed Air.Spring), is accessible through a Google Chrome extension. This allows air.spring to be written purely in Javascript and to be used on any machine that supports Google Chrome. Additional account management features, such as session management and password recovery will also be made accessible through the Air.Auth website.

The server portion (nicknamed Air.Compute), consists of a REST API written in Javascript using Node.js. This configuration supports simple (REST) and high-performance (Node.js) implementations, and allows for the entire project to be written in Javascript. Air.Compute also utilizes a two-tier data management system, consisting of MySQL for persistent storage, and Redis for non-persistent session data.

Progress To Date

A considerable amount of time has been spent setting up and configuring the various project components. To date our team has been able to establish a Node.js application for the REST API, prototype a Google Chrome extension, and integrate various Leap Controller libraries with the Google Chrome extension. The remaining time for this project has been spent writing algorithms for gathering hand data from the Leap Controller, temporarily storing the data in Google Chrome (indexedDB) and using the data to uniquely identify a user's hand.

A complete summary of the work performed on the project to date is as follows:

- Topic discussions and selection. (Includes brainstorming sessions)
- Research on Leap Motion Device API and selection of relevant hand data.
- Setting up a development environment (client and server).
- Creating User Interface design elements for the project.
- Setting up the Google Chrome extension and integrating it with LeapJS (Javascript library).

- Setting up a Node.js application for the REST API.
- Developing the bio-authentication algorithm.

Moving forward, more work is required to perfect the biometric authentication. Once this is complete, gesture recognition can be implemented, the API can be completed, and the frontend user interface can be finalized.

Log Book

Week	User	Task	Progress	Time Spent
1	Mishra			
		(Group Meeting) Topic discussion	Complete	~ 4 hours
	Cole			
		(Group Meeting) Topic discussion	Complete	~ 4 Hours
	Conrad			
		(Group Meeting) Topic discussion	Complete	~ 4 Hours

Week	User	Task	Progress	Time Spent
2	Mishra			
		Reasearch feasibility of Leap Motion	Complete	4 Hours
		(Group Meeting) Project decision	Decided on leap motion hand authentication	2 Hours
	Cole			
		Research possible competition	Complete	4 Hours
		(Group Meeting) Project decision	Decided on leap motion hand authentication	2 Hours
	Conrad			
		Research techniques to uniquely identify hands	Complete	4 Hourse
		(Group Meeting) Project decision	Decided on leap motion hand authentication	2 Hours

Week	User	Task	Progress	Time Spent
3	Mishra			
		UI initial design	Initial user flow complete	4 Hours
		(Group Meeting) Initial implementation of dummy Chrome extension	Dummy Chrome extension running	2 Hours
		(Group Meeting) Progress Report 1	Complete	3 Hours
	Cole			
		UI initial design	Initial user flow complete	4 Hours
		(Group Meeting) Initial implementation of dummy Chrome extension	Dummy Chrome extension running	2 Hours
		(Group Meeting) Progress Report 1	Complete	3 Hours
	Conrad			
		(Group Meeting) Initial implementation of dummy Chrome extension	Dummy Chrome extension running	2 Hours
		(Group Meeting) Progress Report 1	Complete	3 Hours

Week	User	Task	Progress	Time Spent
4	Mishra			
		(Group Meeting) Analysis of data available from Leap Motion	Decided on important data points to gather from Leap Motion	4 Hours
		(Group Meeting) First presentation preparation	Presentation complete	3 Hours
		Presentation	Presentation well received	3 min
	Cole			
		(Group Meeting) Analysis of data available from Leap Motion	Decided on important data points to gather from Leap Motion	4 Hours
		(Group Meeting) First presentation preparation	Presentaiton complete	3 Hours
		Presentation	Presentation well received	3 min
	Conrad			
		(Group Meeting) Analysis of data available from Leap Motion	Decided on important data points to gather from Leap Motion	4 Hours
		(Group Meeting) First presentation preparation	Presentation complete	3 Hours
		Presentation	Presentation well received	3 min

Week	User	Task	Progress	Time Spent
5	Mishra			
		Created the Node.js infrastructure	Running - Some endpoints completed	2.5 Hours
		Created a new Node.js application(Air.Compute)	Complete	1.5 Hours
		Created unique password encryption and decryption	Complete	3 Hours
		Configured MariaDB and Redis for the API	Complete	2.5 Hours
	Cole			
		Configured aircompute development environment	Complete	1 Hour
		Prototyped gesture recording	Initial impression looks promising	3 Hours
		Information gathering and discussion on current API structure	Complete	1 Hour
	Conrad			
		Configured aircompute development environment	Complete	1 Hour
		Created IndexedDB for data storage within Chrome extension	Complete	4 Hours
		Implement Hand data extraction and storage within extension	Complete	3 Hours
		Information gathering and discussion on current API structure	Complete	1 Hour

Week	User	Task	Progress	Time Spent
6	Mishra			
		Created user sign up and API registration endpoints	Complete	0.5 Hour
		Implemented Chrome extension ajax calls to the API end points	Complete	1.5 Hours
		(Group Meeting) Analysis of accuracy of hand dimension	Identified certain measurements that were consistent	3 Hours
		(Group Meeting) Discussion of possible relationships to uniquely identify hands	Identified some relationships - more testing needed to determine accurary	3 Hours
		(Group Meeting) Second presentation preparation	Complete	2 Hours
		Second Presentation	Complete - Ran a little over time	3 min
	Cole			
		(Group Meeting) Analysis of accuracy of hand dimension	Identified certain measurements that were consistent	3 Hours
		(Group Meeting) Discussion of possible relationships to uniquely identify hands	Identified some relationships - more testing needed to determine accurary	3 Hours
		(Group Meeting) Second presentation preparation	Complete	2 Hours
		Second Presentation	Complete - Ran a little over time	3 min
	Conrad			
		Created Dynamic graphs from hand data to view consistency	Complete - Used in presentation	2 Hours
		(Group Meeting) Analysis of accuracy of hand dimension	Identified certain measurements that were consistent	3 Hours
		(Group Meeting) Discussion of possible relationships to uniquely identify hands	Identified some relationships - more testing needed to determine accurary	3 Hours
		(Group Meeting) Second presentation preparation	Complete	2 Hours
		Second Presentation	Complete - Ran a little over time	3 min