Six Degrees Of Separation

1. Motivation
    1.1.   This product is being developed to allow people to connect with other people. The concept of this project comes from the theory of "6 Degrees Of Separation" which is the theory that all people are connected through six people or less. With the recent development of products, such as Facebook, social networking has become easier, but there are still issues in meeting new people. This product will leverage the existing social networking technologies in order to determine how to contact different a person through people that a person is friends with. Currently people have to use a variety of methods to contact a person by either:
        a) Going through contacts within these social networking web sites and finding a path to the person that they wish to contact.
        b) Using resources to find the contact information of the person and contacting the person directly.
    1.2.   This product will promote global collaboration by creating an easier way to find the shortest path to a person through existing contacts.

2. Functionality
    2.1.   Basic Functionality
        a) Provide an efficient backend that leverages RESTful APIs from social networking websites, to find a path to a person through people that a person is friends with.
        b) Provide a web based UI to allow a person to interact with the backend.
            • Selecting and providing authentication to different social networking websites.
            • Allow a person to provide a contact information of the person to contact.
                • Name
                • Country
                • State/Province/Territory
        c) Advanced functionality
            • Allow the contacter to send a message to either the contactee or different people that are part of the path to the person.
                • This will make it easier to contact and communicate with people.
            • Interface with the existing social networking sites in order to allow a person to add them as a new contact.
            • Allow a user to click on different nodes in the path and find an alternate route to the end person.
        d) Other Constraints.
            • UI must be intuitive.
            • Securely transfer this information in order to maintain privacy, of the user.

3. Platform Description
    3.1.   The front end will communicate with the server through the use of AJAX with periodic polling for updates and responses as the logic to create the graph of connections between users could potentially take a while.This platform will leverage HTML5 and canvas(VML for IE) for the UI this will allow for a cross browser

implementation that allows for dynamic interactions and cutting edge interactions without the use of flash.

   3.2.   The back end will be a wrapper for the RESTFul services that are leveraged by our web-app. The server should make and receive requests from the different services, parse and return to the front end in a standard format that is the same for the different social sites. The technology used could include either node.js or clojure depending on the speed of parsing the contacts in order to create the graph.

4. Technical Challenges

   4.1.   Technical Ability
     a) The team is made of different skill levels.
     b) This application is leveraging advanced and new technologies that members of the team may not have used before.

   4.2.   Security
     a) This web app is transferring user information over the web so we will have to leverage HTTPS and certs to handle encryption.
     b) Connection with different APIs will also be a challenge since they use different authentication patterns.

   4.3.   Speed
     a) This web app is doing intense graph creation parsing hundreds of nodes in order to create a graph that the front end can display to the user.
     b) Efficiency on the server will also depend on the different services that are being used since a request will be made for each level of the graph possibly resulting in hundreds of connections.
     c) May want to leverage caching information on the server in order to reduce network calls.

5. Team Composition

   5.1. Douglas Martin
     a) I have worked for different companies creating enterprise solutions for different types of problems, including weather, GIS, and auctions. I have experience in in developing and deploying products leveraging the following technologies :
       • Oracle, Postgres, PostGIS, MySql, SQLServer, Perl, Python, JavaScript(including JQuery, Dojo, YUI, and ExtJS), CSS(2 and 3), HTML 5, Java(certified in both SE and EE), C, C++, ArcSDE, ArcMAP, ArcObjects, COM/DCOM, node.js, Ruby, and many more application frameworks and languages.
     b) My main focus currently is working for a company in Kansas City called Pollenware, which leverages node, ruby, and MySql on the server, and dojo, yui, and raphaelJS on the front end. We use gitHub, JIRA, and Confluence for ticketing, tracking and inner documentation. For our software development we leverage a mixture of agile, waterfall and scrum(for planning of sprints). In previous projects I have used waterfall approach and found it to be slow.

   5.2. Justin Hamilton
     a) My primary experiences in programming are academic as well as building several professional back-ends to server applications and logic systems. My experiences lie primarily in C, C++, Java, Python, Perl, and Lisp.
     b) I currently work for the Air Force Weather Agency working on web services using Java and Python. My current project includes working with meteorological data and a large- scale geographic information system.

   5.3. Josh Wiley

a) I have real-world experience in front-end web GUI, using html, css, javascript, and jquery. I also have experience with php and mysql. At my student worker position, I do a lot of graphic design for publications for the college of IS&T. I am also familiar with Java and C.

5.4. Anna Buhman

a) I have experience writing code at my internship at Union Pacific, where I mainly create software for a user-friendly interface for employees in other departments at Union Pacific to interact with the database. The languages I have used before include Java, C, Perl, Coldfusion, Javascript, SQL, and html.