Section 1

The group decided to research information for Robert Pinto's web service idea Secondhand Knowledge Resource. With an interest in the programming necessary for the design of the web service, Eric Cherin was tasked with investigating the necessary services, languages, encryption, and algorithms. The team leader Vincent Gasbarro utilized aspects from his ongoing research into data visualization and also researched possible backend hosting services for the web service and landing pages. Robert Pinto researched ways methods of making the actual real-world connections between high school and college students.

	Vincent Gasbarro	Eric Cherin	Robert Pinto
Percentage of Effort	34%	33%	33%

Section 2

Before any technical aspects are considered, the concept of building a service must be evaluated. To simply build a service is inefficient as the user doesn't know what they truly want. In the case of Secondhand Knowledge Resource, the intended value of the service is to provide high school students with a channel to obtain information from current college students. However, the method in which that channel is designed is critical if a successful product is desired. While it is possible to build a service that the designers believe is perfect, the lack of user input during the initial stages of design could cause a backlash to the final product. The worst case scenario would be that no users find value in the product thus resulting in a massive failure and time sink. It is thus in the group's best interest to use a build-measure-learn feedback loop. By building just the necessary components of an idea, a minimum viable product, the group can observe if potential users see value in the proposed idea. From there the group can obtain a general picture on features a potential user desires. Finally, an updated product can be built and the loop can continue refining based on user reactions. This process would greatly expedite the creation of a successful service. [1] The remainder of this report will report on both the basic technical necessities that a MVP would require and common technical details that are used in existing successful web services.

Hosting Service

The most important component of a web service is the actual hosting platform that allows a user to access the product through the internet. In terms of the best free web service platform, Amazon's Elastic Compute Cloud (Amazon EC2) is a top candidate. Amazon EC2 allows a developer to choose a pre-configured Amazon Machine Image (AMI) and through a very simple interface, have a fully functional Linux virtual machine available to host a web service on. The EC2 virtual machine is secured through the use of key pairs that use the authentication function of SSH within puTTy. puTTy is a free telnet/ssh client that allows for emulation of a Linux terminal on a Windows machine. It is a key program that allows for access to the personal Amazon EC2 from any computer with the authorized key pair. From there, knowledge of Linux allows the installation of a LAMP (Linux-Apache-MySQL-PHP) server to serve as the backbone of the web service. The Amazon EC2 free tier allows for 750 hours using the EC2 Micro instance which is more than sufficient for the scope of the project [2]. Another requirement of

hosting is the purchase of a domain name. This step does cost money, but many websites such as godaddy.com allow for the purchase of relatively cheap domain names ($15 for a year) [3]. Once purchased, the domain name's zone file can be edited in conjunction with the apache server's configuration file and allow for a simple html landing page to be hosted. With the completion of this process, the backbone of the MVP is ready to be launched as a simple html landing page expressing what value the potential product is providing can be used to garner interest. The feedback, whether positive of negative can then be used to determine what the next step should be.

Another important development component is deciding what web application framework to use. A web application framework supports the development of dynamic websites, web services, and web resources. It is based on the model-view-controller (MVC) software pattern, a process loop that defines interactions between key aspects of web applications. Broken down, a user uses the controller which then sends commands to the model. The model recognizes this change in state from input and updates the view that the user is seeing [4]. The two web application frameworks researched are CakePHP and Django. These frameworks both follow the MVC approach but vary in programming languages, PHP and python respectively. They both are fully supported on a Linux environment and have friendly licenses that are perfect for poor college students that can't afford costly services. Both frameworks also have a plethora of documentation on their respective websites to allow for easy learning and use cases. Between the two, Django is more well-known as it is being utilized by popular services including Pinterest, Instagram, and Mozilla. However, the final decision regarding which service to use depends on the group's proficiency with PHP or python [5] [6].

Visualization

The way information is visualized is an important concept to keep in mind for a service that intends to be an information source. The style in which information is presented, whether it is straight facts or embellished to be more artistic can have an impact on how the user gains insight. The work of Moere et al. [7] conducted experiments to determine the differing impact between three visualization styles on this hypothesis. A key conclusion reached from the experiment was the over embellishing a visualization with visual or interactive features is counterproductive if usability is an important concern. For the group, this finding would best be tested using the build-measure-learn feedback loop. By introducing varying levels of interaction and using A/B tests, the group can determine the best mix of features to satisfy users. Another key finding was that by limiting interactive capabilities, users can be steered towards specific behaviors. This finding could be useful if we find that users are having trouble accessing desirable resources. Again through testing, the service could improve the matching and interaction of high school students to college students.

Making the Connections

While there are many aspects of this service that will need to come together for it to function, the one that will be the absolute foundation for it will be the connection between the high school students and the college students. This means that every detail of this connection must be completely analyzed and planned out, from how they will be found and created to how the first interactions will occur. To begin, we took a step back and searched for any research or funded projects that may have been related to our service.

It seems as though there has not been anything done with regard to connecting high school students to college students as our group plans to do. There have, however, been a number of NSF grants given out to groups who attempted to implement types of peer mentoring programs that would promote and improve STEM learning. This fact opened our eyes to the possibility of a zoom-in pivot for our project. It is no secret that the US has "struggled to persuade sufficient number of its citizens to pursue highly technical careers,"[8] so it is possible that by connecting college students of the STEM disciplines with high school students we could increase the number of people entering these fields of study. This would be taking our original idea and using it as a possible resolution for this clear deficiency. Right now, if high school students in the United States do not attend a school that affords them to opportunity to explore these disciplines, then there is no real draw for the students to enter into them after high school. This service has the potential to give all high school students this opportunity by allowing them to connect and interact with college students who have made that decision.

Moving on, we discovered many techniques and features that could be implemented in our service. The next few paragraphs will discuss these and how they might fit into our original project, that is, the service without the aforementioned zoom-in pivot.

A big question for this project has always been "How will the connections be made?" because there are a fair number of possibilities. After much deliberation, it seemed as though the best course of action would be to look at our project as though it was a dating service for the purpose of finding out ways to connect the users. As with many of these services, one possibility could be to survey the users for basic information and interests and then give them a number of matches and profiles to browse. Then in the future, suggestions could be made based on previous selections. *A Survey of Explanations in Recommender systems* (http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4401070&tag=1) gives some great insight into some of the very popular recommender systems.

Another option for making connections is to take an approach similar to that of the service called OKCupid. For this service, users are not only asked questions, but they are asked to weigh the importance of the questions as well. This helps to match users in a special way that is not available through only surveying.

A third possibility, which could also be implemented alongside one of the others, is a search engine for the high school students to use to look for available mentors in any areas. This is beneficial because if the service is expanded to the point where there are a number of different categories (fitness, work, knowledge, future schooling…etc.), then the users will be able to find more than one peer to fulfill their needs. This would mean that the relationship between the high school and college students would most likely be many to many instead of one to one. *People Searching for People: Analysis of a People Search Engine Log* (http://delivery.acm.org/10.1145/2010000/2009927/p45-weerkamp.pdf?ip=155.246.116.214&id=2009927&acc=ACTIVE%20SERVICE&key=7777116298C9657D%2EC5AD4D952AEC6BE6%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=298454950&CFTOKEN=28821817&__acm__=1393925211_3fba6f36fe7bb3faf52b80af3a8e9db5) provides some great information on how searching for people on the internet works and what the recent trends with it have been.

Once the connection is made, the process is far from over. First interactions must be handled with care because they have the power to easily dictate how a person feels about the service as a whole. For that reason, a number of processes and techniques will be examined to determine what will be the best fit for our project.

To ensure a good first interaction and in turn lock in a customer, the first thing that needs to be perfected is the set up. After researching and examining approaches by other companies, something that really stood out was the progressive commitment approach taken by TravelRole.

Their intention was to connect individuals who were looking to carpool and accomplish something else during the commute. In their words, “the system matches people to rides that allow them to link carpooling with a personal goal such as practicing their French, talking about rebuilding an engine, discussing religion, and so on.”[9] It seems as though it would be almost impossible to make these connections, but the way that their technique slowly draws in users makes it attainable. For our service, it could mean that we allow users to view their matches before joining, or to search and establish some connections first and then slowly introduce them to the entire service.

Development

As discussed before, the goal of this project is to connect curious high school students to knowledgeable college students. In order to accomplish this, the group needs to design an interface to connect students to the service. By looking at other successful websites, the interface must be intuitive, be able to easily exchange files, and have the ability to support both video and text chat communication. These tasks can be accomplished by using Google Web Designer, which incorporates HTML5 and CSS3. Google Web Designer is used to create text and 3-D objects that can be animated. It can output the source documents in HTML5, CSS3, and JavaScript. It also serves as a text editor to further assist the programmer. HTML5 is a markup language used for displaying content on the internet. It has multi-platform support that can be used to display multimedia and graphical content. HTML5 can be used to access JavaScript programs. JavaScript is a programming language that allows scripts to interact with the user, dynamically change content, and control the browser. For example, JavaScript will enable the group to incorporate outside services such as Skype and PayPal. CSS3 is a style sheet language that is used to describe the formatting and layout of a document such as a webpage. It lets the group change font properties such as color and type of font in a markup language such as HTML5. By using the Google Web Designer, these languages will enable the group to design a GUI, video files, and interactive graphics. The use of Google Web Designer also opens another web hosting service display the group’s created content on the internet. Google Sites is a viable web hosting service because it is compatible with Google Web Designer [10] [11] [12].

The web service will need to handle information about the user. Important functions necessary include storing and retrieving usernames, passwords, and personal data. This can be accomplished by using a database to capture, organize, and retrieve data. There are two major types of databases: SQL and NoSQL. SQL is a relational database which uses a set of tables that store data in categories. Each table consists of categories in columns and rows contain an instance of the data that has been defined in the columns. Everything must be predefined because structure and data types are fixed in advance. NoSQL is not a table based database because it just uses a key to access a certain value. The value can be documents or data. This allows freedom because structure and data types are dynamic. If we would like to implement an SQL database we can use MySQL or if we would like to implement a NoSQL database we can use MongoDB [13].

In order to facilitate searches, algorithms must be investigated. A key search algorithm is how to define a matchmaking algorithm to pair college students and high school students together. Weighing factors like type of college and type of degree to determine the best pick for match is a possible method. To make the user’s information secure, the web service must be able to encrypt

passwords. One way to accomplish this is to hash and salt passwords using an algorithm such as SHA-2 or Bcrypt. The algorithms take in the password, a key, and salt to generate a hashed and salted password. Hashing the passwords prevents the actual plaintext password from ever being secured. Instead, the hashed and salted password is stored in the database. To verify the user, the user first inputs the password. Next, the algorithm computes the hashed password and compares the result against the result stored in the database.

References

[1] Eric Ries. *The Lean Startup.*

[2] Amazon EC2: Sourced from: http://aws.amazon.com/ec2/?sc_channel=PS&sc_campaign=AWS_Free_Tier_2013&sc_country=US&sc_publisher=Google&sc_medium=Brand_Core_EC2_E&sc_content=36175459002&sc_detail=Amazon%20ec2&sc_category=compute_networking&sc_segment=compute_networking&sc_matchtype=e

[3] GoDaddy. Sourced from www.godaddy.com

[4] Buschmann, Frank (1996) *Pattern-Oriented Software Architecture*.

[5] CakePHP. Sourced from: www.cakephp.org

[6] Meet Django: Sourced from www.djangoproject.com

[7] A. Moere, M. Tomitsch, C. Wimmer, C. Boesch, and T. Grechenig. Evaluating the Effect of Style in Information Visualization. In IEEE Transactions on Visualization and Computer graphics, volume 18, pages 2739-2748, no.12, 2012.

[8] “Losing the Competitive Advantage? The Challenge for Science and Technology in the United States,” [Online document], p. 4, February 2005, available HTTP: http://www.aeanet.org/publications/IDJJ_AeA_Competitiveness.asp

[9] Selker, T., “TravelRole: A Carpooling / Physical Social Network Creator” Carnegie Mellon Univ., Moffett Field, CA, USA ; Saphir, P.H. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5478453&tag=1

[10] https://www.google.com/webdesigner/

[11] http://developer.skype.com/skype-uris

[12] https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5

[13] http://www.mongodb.com/