**Introduction**

Often, college students run into particular courses or topics in which they struggle; however, they are frequently also strong in other areas. If there were a system in place for students to learn others’ strengths and weaknesses, it would be much easier for students to meet those that can help them and those that they can help. This is what our application would do. A student could log on and specify the courses and topics in which they are weak and be informed of any students that are capable and willing to lend a hand to them, or vice versa. This would work extremely well for any direct matches, where two students can help each other in different topics; however, it can also be used to set up group tutoring and collaboration.

There are a few challenges faced in creating a tutoring system for college students. The first is to get enough people to make the application worth using. Since it’s a social network of sorts, Metcalfe’s Law applies, and the value of the application increases at an exponential rate as new users join. For our application, this is a correlation to the actual function of the site. The more people join, the more likely it is to find a match for the user. If the application wasn’t liked quickly from the start, it would most likely fail. Thus, the application has to be in a final, entirely usable form before being released. This would most likely involve plenty of user testing in order to be sure the average college student at RPI would enjoy using it. Another is to create a quick and easy to use application, as most students don’t want to spend too much time on things such as this. Their time is devoted to their projects and work, and their social life. This application is meant to give more time back to those activities, not taking from them. Thus, all information the user needs should be and is presented in a compact and easy to use format, and there is no excess. Another challenge is to create a database in a way that reliably and quickly gives all sorts of matches to the user. Again, the application needs to be fully usable from the start, and this means that the back end needs to be entirely functional and easy to edit and draw from.

**User Interface**(See *IANav.pdf* for Information Architecture for Navigation)

The first time a user goes on our site, he or she would have to make a profile. To create their profile initially, the user would have to enter their first name, last name, RCS login, and a password of their choosing. They are then redirected to the “Edit My Profile” page. On that page, they can put in their strengths, weaknesses, contact info, and they have the option to change their password. The user can return to this page at any time to add to their strengths and weaknesses or to change some of their information. If they already have a profile, they just have to enter their RCS login and password to enter the site.

After the user has logged in, when the user presses the logo in the upper left corner, they would be directed to the logged-in home page. This page has three sections: Perfect Match, People Who Can Help You, and People You Can Help. Each of these sections has lists of people’s names, along with their major and the subjects they can help in and need help in. The Perfect Match section is a listing of people who need help in the subjects you can help in and can help in the subjects you need help in. In the Perfect Match section, when the user clicks on one of the people’s names, an alert box comes up with their contact information. The same applies to the People Who Can Help You and the People You Can Help sections. The user can then contact the person themselves, using the information given.

When the user clicks on the heading of either the People Who Can Help You or the People You Can Help sections they are redirected to a page dedicated to either people who can help them or people they can help, depending on which one they clicked. This page will allow them to search within the results by course type, level of course, and by keywords. From this page, they can find people who fit their specifications.

**Nielsen’s Ten Usability Heuristics**

**Visibility of System Status**

Users will be informed of the current status of and changes to the system by the information box on the front page of the web site. The box will contain information pertaining to recent changes, new features, and any planned downtime of the application.

**Match between System and the Real World**

Firstly, the application is being developed by college students for college students; the target audience is one that the developers are intimately familiar with. The application's data is presented on a series of contextually specific pages, with each page maintaining a sectioned format that separates relevant information by category.

**User Control and Freedom**

Navigation within the application is built to be easy and intuitive, with a consistent header on each page of the application that provides navigation back to the “Home” page. For example, detailed information can be obtained by clicking on a summary heading. Data is entirely user-provided and is always able to be updated by the user.

**Consistency and Standards**

Titles and information are presented on each page in a consistent and easy-to-understand format.

**Error Prevention**

The forms used to supply information are validated before submission to help prevent user error. For example, auto-completion of form elements is used in places where data can be repeated.

**Recognition Rather than Recall**

The interface for the application is designed to be intuitive and easy. For example, links are clearly labeled and available on each page.

**Flexibility and Efficiency of Use**

Auto-completion is available in the data-entry parts of the application. This makes it easier for users to get results from the application with less time spent entering data. Simplified bits of data are available on the user's home page so that important data can be communicated in a clear and concise manner. For more detailed information, the user can click on the section and get more in-depth information.

**Aesthetic and Minimalist Design**

Only relevant information is collected and displayed to and from the user. For example, on the home page, users are able to access data in one central location without additional interaction.

**Help Users Recognize, Diagnose, and Recover from Errors**

Form validation, including detailed error messages, helps users identify and fix problems during the data-entry process.

**Help and Documentation**

Contextual help is available on each page to help direct the user to requested information.

**Data Model**(See *IADB.pdf* for Information Architecture for Database Access)

The data model for this application is heavily reliant on relational database technology. A relational database enables the data in our application to be separated into manageable sections while also retaining the ability to easily access interrelated records. Because our application uses only one main set of data, our schema will contain one database which will consist of multiple related tables. The main table will contain the core data for each user of the application. Each user will have one record in the main table which consists of the user's name, a unique identification number, a text field containing the user's contact information, and authentication data (an ID and an MD5 password hash). The database will contain two additional tables: one for storing the strengths of the user and one for storing the weaknesses of the user. Each table will consist of one record per strength or weakness with two fields per record: one to store the strength or weakness and one to store the unique identification number of the user to which the strength or weakness corresponds.

The contact information for the user can be accessed by selecting the user's record from the main table by unique identification number. The authentication information will be validated by hashing a supplied password and then querying the main table for the user record which contains the supplied user name and hashed password; one record as a result would indicate a correct user name and password, while zero records as a result would indicate incorrect authentication data. A user's strengths and weaknesses can be accessed by querying the strengths and weaknesses tables for strengths and weaknesses whose records contain a user's unique identification number that corresponds to the current user's unique identification number. This unique ID will be stored in the user's session so that any data needed to render a page within the application can be accessed using the unique ID of the current user.

As an application centered on user information and personal data, the data will be entirely user-supplied. Upon creating a new account within the application, the user will be asked to supply strength and weakness data. All information will be collected via web form and subsequently stored in the proper database tables with an insert statement. The user will also be able to change supplied data via web form; changed data will be updated in the database with an update statement. Interface elements displaying and accessing data relating to user contact information or authentication information will use data from the database's main table, while elements displaying and accessing data relating to the user's strengths and weaknesses will use data from the database's strengths and weaknesses tables. For example, the “Perfect Match” interface element will rely on a database query that takes the currently logged-in user's strengths and queries the database for users who display these strengths as weaknesses, and that takes the user's weaknesses and queries for users who display these weaknesses as strengths. With these two result sets, the application will choose all users who appear in both tables; that is, the application will populate the “Perfect Match” element with users who have complementary strengths and weaknesses to the currently logged-in user.

**Use Cases**

**Use Case 1: Gertie**(See *UseCase1.pdf* for Storyboard)

* Returning User
* Freshman
* Biomedical Engineering
* Excellent in IEA and CAD
* Needs help in Physics 1 occasionally

Gertie finds some free time on a Wednesday afternoon. There's a IEA exam approaching at the end of the week and a Physics exam approaching the following week. Gertie decides to log on to our application to see who she can help to study for IEA. She enters her information in the log-in box (Figure 1) and is presented with the home page. Gertie checks her Perfect Match box and notices that a new student has registered that is a Perfect Match with Gertie: Ernie desperately needs help with IEA, but has excelled in Physics I. Ernie's information appears in Gertie's Perfect Matches section (Figure 2). Gertie clicks on Ernie's “Contact” button and is presented with a link to Ernie's Facebook page. Gertie requests to be friends with Ernie on Facebook and then returns to the application’s Home Page. There, Gertie checks to see who she can she help with CAD and finds three new users to help (Figure 3). She only finds two people to help with CAD, but doesn't recognize their names from her section, so she clicks on the section header to get more information on people she is able to help. Gertie is presented with a list of users that she is able to help and clicks on the “Subject” column header to sort by course so that she is easily able to see the people she can help with CAD (Figure 4). From there, Gertie can find contact information for any user with CAD listed as a subject with which they are in need of assistance.

**Use Case 2: Boris Castillo**(See *UseCase2.pdf* for Storyboard)

* New User
* Junior
* Aeronautical Engineering
* Desperately seeking help in MANE-4070 (Aerodynamics 1)

Boris notices half way through the semester that Aerodynamics 1 is having a terrible effect on his GPA. He has tried studying more, but always finds that he works better when guided by someone who can explain the underlying concepts effectively. Boris decides to seek out a tutor that can provide assistance with Aerodynamics 1 so that he might be able to pass the course. He asks around and hears that our application might be able to help him find someone who can help. He browses to the web site, reads the application description, and sees that it might indeed be helpful to him (Figure 1). He enters his first name and RCS ID in the sign-up form on the front page (Figure 2) and attempts to submit the form. Because he has not provided all required information, the form fails validation and prompts him to complete the form before proceeding (Figure 3). He enters the required information and again attempts to submit the form. The form is submitted and the system creates Boris a new account; Boris is then directed to a page that asks him to provide information that can match him to other users that might be able to help him (Figure 4). He specifies that he would like assistance with Aerodynamics, but does not think that he would be helpful to others, so he does not provide any courses with which he is able to provide assistance (Figure 5). He submits this form and is then directed to his home page, on which he is able to identify the top three users who can help with Aerodynamics (Figure 6). He clicks on the section header and is presented with a page on which he can find all users that can help with Aerodynamics. He sorts the list by class to find other Juniors who might be able to help him (Figure 7). Boris is now able to click on the users' names and collect their contact information to pursue help with Aerodynamics.

**Prototypes**

See *LoFi.pdf* for Low-Fidelity Prototype and *MedFi.pdf* for Medium-Fidelity Prototype