

Metis

By

Matthew A. Huschart

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

May 2002

Metis

By

Matthew A. Huschart

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

© Copyright 2002 Matthew A. Huschart

The author grants to the Information Engineering Technology Program permission to reproduce and distribute copies of this document in whole or in part.

Matthew A. Huschart

Date

Prof. Russell McMahon, Faculty Advisor

Date

Lawrence G. Gilligan, Department Head

Date

Acknowledgements/Dedication

I would like to thank Dr. Geonetta, Prof. McMahon, Prof. Schlemmer, and the rest of the IET faculty at the OMI College of Applied Science for not only their teaching and support in the Senior Design sequence, but also their dedication to the students over the past three years.

I dedicate this project to my wife, Molly, and thank her for all her support, understanding, and encouragement during my pursuit of a Bachelor of Science degree in Information Engineering Technology.

Table of Contents

Section	Page
Acknowledgements/Dedication	i
Table of Contents	ii
List of Illustrations	iii
Abstract	v
1. Statement of the Problem	1
1.1. Definition of the Need	1
1.2. Current Offerings	2
1.3. How Humans Learn	4
2. Solution	5
2.1. Initial Areas of Information	9
2.2. User Profile	10
2.2.1. Student Profile	10
2.2.2. Teacher Profile	10
2.3. Flowchart and Navigation	11
2.4. Screen Description	12
2.4.1. Welcome Screen	12
2.4.2. Login Screen	12
2.4.3. Create Login Screen	13
2.4.4. Failed Login Screen	14
2.4.5. Customize Menu Screen	15
2.4.6. Build a Classroom	16
2.4.7. Profile Information / Class Information	17
2.4.8. Flashcards	18
2.4.9. Reports	19
2.4.10. Logout Page	20
2.5. Database Design	20
2.6. Code	21
2.6.1. Visual Basic	21
2.6.2. Active Server Pages	21
2.6.3. Java Script	21
2.6.4. General	22
2.7. Color Scheme	22
3. Objectives of the Project	22
4. Proof of Design	23
5. Conclusions	24
6. Recommendations	26

Appendix A: Timeline	28
Appendix B: Budget	29
Appendix C: Database Relationships and Table Descriptions	30
Appendix D: Proposed Testing	36
Notes	39
References	40

List of Figures

Figure 1. Summary of current learning Web sites	3
Figure 2. Three learning stimuli	5
Figure 3. Initial information areas	9
Figure 4. Metis flowchart and navigation for user interface	11
Figure 5. Welcome screen	12
Figure 6. User login screen.	13
Figure 7. Create login screen	14
Figure 8. Failed or invalid login screen	15
Figure 9. A student's customized menu screen	16
Figure 10. Build a classroom	17
Figure 11. Student's profile information / class information	18
Figure 12. Example flashcard screen	19
Figure 13. Student report	19
Figure 14. Logout page	20
Figure 15. Database relationships (1)	30
Figure 16. Database relationships (2)	31

Abstract

The purpose of Metis is to provide students with "flashcards" that mirror their classroom studies. Metis was designed to work as a preparatory test partner. Students will have access to topics specified by their teachers and they can test their knowledge of these topics by answering questions. A database of questions was created with selected topics: Metis reads and randomly generates questions for the users. Users have access to reports that show their overall progress and their progress as compared to their classmates.

Metis is different than many of the current learning Web sites in numerous ways. This presentation discusses the problems with the current Web sites and the advantages of Metis. It describes the developed solution from idea composition to finished product.

Metis

1. Statement of the Problem

1.1 Definition of the Problem

It is hard to participate in any activity today and not use or be influenced by a computer in some way. A computer under the hood of a car is taking care of things like calculating the gas mileage and determining the speed or direction of travel. Some professional baseball players, Curt Schilling for example, track every at-bat that opposing hitters have had against them, down to the pitch level, using computers and databases. Some countries are even starting to test the use of cell phones to get items from vending machines; the \$.50 charge just shows up on the phone bill.

The Internet is also everywhere. Products and advertisements have Web site addresses listed on them. People conduct their Christmas shopping from their homes. Schoolteachers supply handouts and assignments to their students through Web portals like Blackboard, used at the University of Cincinnati. People can get up-to-the-minute news and sports scores almost anytime. Just about any topic can be researched and within seconds megabytes of information are at one's fingertips.

I am a very poor speller and my abilities are actually degrading because of the computer. I create a document or spreadsheet at work and rely on the spellchecker to verify my spelling. When I work in an environment that does not have spellchecking abilities, like in UNIX or a programming environment, my signature is all over my work. Misspelled words everywhere. Sadly, this has become a joke in my office.

In the spring of 2001 I formed the idea of making the computer help me become a better speller. I thought back to when I was in grade school and learning how to spell.

My father would ask me to spell the words. Spell "cat", he would say. I would reply "C-A-T". He would then say "elephant" and I would reply "D-O, no dog is supposed to be the next word". After being sent back to my room to study more, I quickly learned that you have to mix things up when learning because they may not be in that order in the real world. Computers can easily "mix things up" and so I thought this might be a good idea. The idea quickly evolved into not necessarily making me a better speller, but could I do something to help grade school students learn and work in more areas than just spelling.

1.2 Current Offerings

I started by researching what is currently available. Not having to look too far, I started noticing advertisements for LeapFrog. This is a system of devices that help children with phonics, reading, and spelling. In 1995 a California lawyer, Mike Wood, was looking for a product that would help his son with his phonics. He was so disappointed with the current offerings that he partnered with people who had education backgrounds and started the company. Six years later, it has become a very successful company with one intriguing twist. The products can continue to grow with the children's learning. When the child has become proficient at spelling the words that came with the speller, the parent can go to the company's Web site and download more words. I liked this idea of a product being able to grow with the learner but I knew that creating a product of this complexity was not what I had in mind.

I decided to narrow my search and look at only current products that are available for a computer. This meant either a CD-ROM or the Web. The CD-ROM is a useful medium for learners; it can be interactive, have animation, color, and sound. The range of information that CD-ROMs cover is extensive; from something as introductory as learning to read or simple mathematics to more advanced topics like the M-CAT^A.

The other medium I was interested in was the Web. I checked The Top 100^B for a review of Web sites that fall into the "Education and Reference" category. Fifteen percent of the top Web sites are in this category, but they do not concentrate on any one topic or aid in learning; they are more for reference. The highest rated site, Learn2.com, was described as offering “nifty tutorials on everything from tying basic knots and choosing the right tent to avoiding junk mail and removing a tick. A brilliant site, with colorful cartoons to accompany the lucid prose” (11). There is not much there to help a fourth grader with his or her spelling or multiplication tables.

I next tried to find what teachers and parents thought about the current options and found four reoccurring themes:

1. Why are there so few good free Web sites that help students learn (4)?
2. When taking a deep look into some of the software that students are using in today’s classrooms, one might find that “much of it is little more than electronic games dressed up to look like math or language instruction” (1).
3. The subject matter may not match the classroom subject matter 100% of the time (4).
4. Is there something that can help students pass state proficiency exams (8)?

Sensing this negative attitude about the current offerings, I reviewed some Web sites and here are brief descriptions of some of the sites available to help students learn.

Web Site	Description
www.edu4kids.com	<ul style="list-style-type: none"> • This is probably the best site currently available but it is having financial trouble and may not be around next year. It needs to raise \$6,000.00 before the end of this year to keep going. • It has flashcards or drills for Mathematics, Geography, Presidents, Vocabulary, and Chemistry.
www.aplusmath.com	<ul style="list-style-type: none"> • Only mathematics covered at this Web site. • Does not work with WebTV or Macintosh computers. • Does have the ability to create printable flashcards.

Web Site	Description
www.chemistry.ohio-state.edu/organic/flashcards	<ul style="list-style-type: none"> • Very sophisticated. Gives chemical compounds and the user needs to give the next logical reaction. Very nice but the user does not enter anything, it is all done in their head or on a piece of paper and the correct answer is displayed on the screen.
www.allmath.com	<ul style="list-style-type: none"> • Only mathematics covered at this Web site. • Does not work with WebTV or Macintosh computers. • Site layout causes the user to scroll to answer each flash card.
www.flashcardsonline.com	<ul style="list-style-type: none"> • This is not actually an online flashcard site. The user must order a set of flashcards and install special software on their computer.

Figure 1. Summary of current learning Web sites

1.3 How Humans Learn

Learning is defined as “any relatively permanent change in behavior brought about through experience” (7). A stimulus is “any aspect of the outside world that directly influences our behavior or conscience experience” (7). The standard three learning stimuli are lectures, readings, and labs (5). Babies learn to speak by listening to their parents talk and some students learn best by listening to lectures. These two examples fall under the first learning stimulus. The second stimulus for learning is reading and writing. Even today, I like to take notes from a college textbook rather than use a highlighter because I learn well by doing this. The last stimulus is experience. The three-year-old may be told not to touch the hot grill, but may not really understand why. One touch and the three-year-old learns. This reason is why students go to laboratories and experiment.

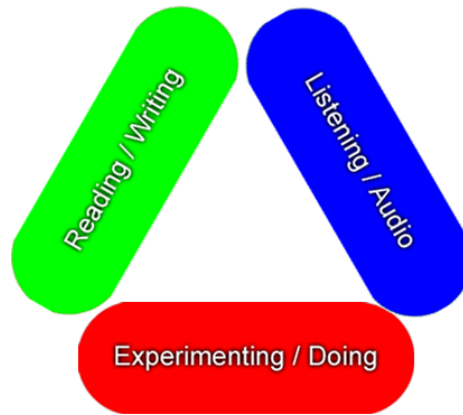


Figure 2. Three learning stimuli

Another important part of learning is repetition. The piano player plays the same melody over and over until it is perfect. The professional basketball player shoots 100 free throws everyday. These people are training themselves to be perfect in specific instances. The student needs to do this, but often not in a specific instance. They can not create a Pavlovian^C mindset for themselves. That is, when they have a question about George Bush, they can not immediately think "Ronald Reagan's Vice President". This would be correct if the question was "Who was President Ronald Reagan's vice president?" but what if the question was "Who was the vice president under President George Bush?" This may not seem that important, but I have personally experienced this. One of my teachers would give quizzes made up of the end of chapter questions. The questions would be ordered differently than in the book but the responses would be in the same order. As soon as I read the question, I would know that the answer was "C". I did not even need to look to see what answer "C" actually was.

2. Solution

Taking all the research into account, I proposed a solution that improved on the inadequacies of the current options and also replicated some of their better qualities.

Specifically, the solution is available at a low cost to the end user. The solution has different areas of information and has enough information in each area so that a user is not getting repeated questions after relatively few questions. The questions are randomized and almost the complete series of questions asked before repeating questions. This works similarly to the way flashcards work; the user goes through all the flashcards in the stack, shuffles the stack, and then starts again. The solution is also aware of the learning stimuli and works to integrate them. Simply put, the solution is visual, audible, and interactive.

The solution could have been deployed in one of two ways, either through a CD-ROM or a Web site. The CD-ROM would increase distribution costs and time. It would also be more difficult to support, as software would have to be installed on users' computers. The Web allows updates to be implemented anytime, at minimum or no cost for distribution, and most importantly, with no support or installation problems. If used correctly, the Web can be accessed from any connected PC or Macintosh computer. It is visual and allows the student to answer the question in multiple ways, such as selecting from a list or typing. It can even speak to the student and act as a 24/7 teacher. These reasons have made a Web site the clear choice for this product. To minimize support requirements, standard HTML^D coding was used. This coding is backwardly compatible with mainstream browsers. For this application, today's mainstream browsers will be defined as Microsoft Internet Explorer^E and Netscape Navigator.

Users of this site will be required to login with school id, login name, and password. These accounts will be setup ahead of time by an administrator. The login pages consist of ASP^F pages written in either Visual Basic Script or Java Script. The

pages will accept the login information and either grant access or display failed login information and help. The login pages have three purposes. First, to limit the number of users that have access to the site. This is not saying that these users will have to purchase a login, rather it is meant to limit the amount of traffic on the site and keep the site from slowing down. Second, the screens will be tailored for what the student is studying: a 4th grader interested in practicing multiplication tables should not have a way to get into the “flashcards” for the Periodic Table because they would not be studying the Periodic Table in their classes. The third reason for the login will be to track the usage of the site. Reports will be created for teachers that show which students have been using this tool and in what areas they have been most active.

The main driver of this solution is a CGI^G program written with Microsoft's Visual Basic. This program accepts input from the user and "reacts" by creating a unique or dynamic Web page for that user. The Visual Basic program is the direct link to the database that stores all the information that students can access. The program generates the pages that can consist of simply HTML code or complex code containing Java Script, such as mathematical pages. For non-mathematical areas of information, the program will randomly select a question for the current area, retrieve the proper answer and three other incorrect answers, randomize the answer order, and format for the user. User One might get this question:

What is the capital of Ohio? A. Cincinnati B. Toledo C. Columbus D. Dayton

User Two might get this question:

What is the capital of Ohio? A. Columbus B. Lexington C. Dayton D. Cleveland

As can be seen, the correct answer moved from "C" to "A" and the incorrect answers changed values or in this case, city names. This will specifically address the Pavlovian learning problem. Users should also not get the same question until they have completed at least 90% of the questions in that area. This means that the users above would not be asked to select the capitol of Ohio again until they have been asked the capitol of 45 of the 50 states.

The database selected for this project is Microsoft Access 2000. A database like Microsoft's SQL Server 2000 is more powerful, but the power would not be used and would increase the cost of the project. The database is the most difficult part of the project because it needs to be designed in a way that can adapt to meet the needs of the different types or areas of information. For example, one of the first areas of information that was developed is "American Presidents". The CGI program will generate questions like "Who was President Washington's Vice President?" It will also generate questions for his years in office, where he is from, and what number president he was.

Another information area to be initially developed was the US States. Users are asked to select the state capitol or state abbreviation. With these examples, the area of the Presidents have four different areas of information that can be prompted for; the area of the states only has two. The database needed to be able to store this information in a way that it is all linked but that does not waste space by reserving space or memory for items that will never be used. So having a table with four fields would be a waste because only two of them would be used for the states. Also, developing a database with tables that way would have had problems because not all information needs to be stored

the same way. Text strings, like city names, are stored differently than numerical values, like what number president someone was.

The last area of the solution, but not necessarily the last to be developed or activated, was the mathematical area. This area works mostly on the user's computer and does not contact the CGI program or the database for information after each question or problem. The numbers to add up are not be stored in the database but are randomly generated; because of this, there is no reason for the program to be communicating with the Web server. This area of the site was written using Java Script. The Java Script page is downloaded and then random numbers are displayed and answers are checked there. The fact that the information will only be downloaded one time^H will speed up this part of the site considerably.

2.1 Initial Areas of Information

The initial areas of information that Metis^I has available for student use are shown in Figure 3.

2.2 User Profile

There are two distinct user profiles for Metis: Students and Teachers.

2.2.1 Student Profile

The students range in age from fourth graders through seniors in high school. Their IT literacy will vary as much as their age. Many of the younger users will not have much computer experience beyond the gaming world. On the other hand, the older students should have taken at least one computer class in high school and would be considered as having moderate computer or IT skills.

Information Area	Example Questions / Flashcards
American Presidents	<ul style="list-style-type: none"> • What number president were they? • Who was their Vice President? • Where are they from? • What years did they serve? • When were they born or died? • How did they die?
US States	<ul style="list-style-type: none"> • What is the capitol city? • When did they join the Union? • What is their abbreviation?
Periodic Table (Elements)	<ul style="list-style-type: none"> • What is the abbreviation or atomic Symbol? • What is the atomic number? • What is its atomic mass? • What is its boiling point? • What is its freezing point?
Vocabulary	<ul style="list-style-type: none"> • The word will be said and users must type it in. • Which of there are synonyms for **** ? • Which of there are antonyms for **** ?
Mathematics	<ul style="list-style-type: none"> • Addition problems.** • Subtraction problems.** • Division problems. • Multiplication problems.** • Rounding Problems. • Converting numerical values into words (like when writing a check). <p>**Users will be able to select the number of numbers to work with and their minimum and maximum size.</p>

Figure 3. Initial information areas

2.2.2 Teacher Profile

The teachers group also has a wide range of IT literacy. Some teachers have been keeping students' grades with spreadsheets for years; others are reluctantly giving up their paper grade books because they either do not understand spreadsheets or because they are afraid of using a computer. In my project's initial research, I found that some schools,

like Oak Hills High School, were forcing their teachers to start to use the computer in the class room. For this current year, all the teacher had to do was create a simply static web page. It could be done in about 10 minutes with Microsoft Word and exporting to an HTML file, but these are the small steps that schools are trying to take to get all teachers more comfortable with computers.

2.3 Flowchart and Navigation

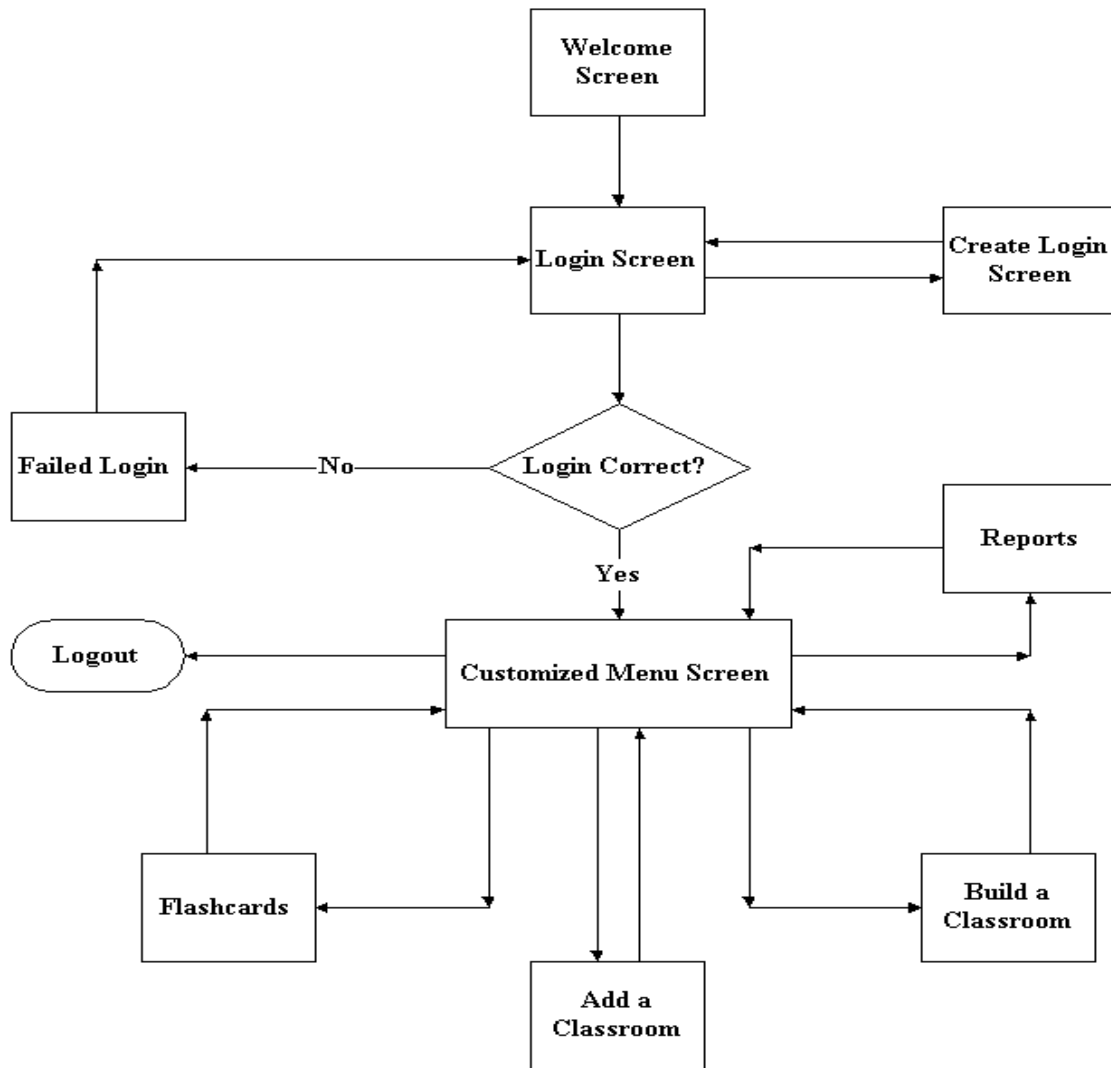


Figure 4. Metis flowchart and navigation for user interface

2.4 Screen Description

2.4.1 Welcome Screen

The Welcome screen consists of a Flash movie and a link. The link allows the user to go to the login screen at any time. The entire Flash presentation does not have to be seen.

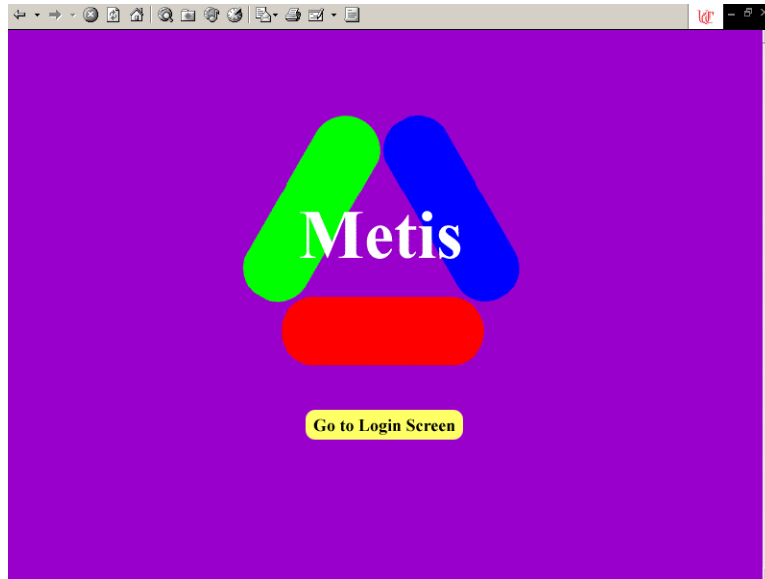


Figure 5. Welcome screen

2.4.2 Login Screen

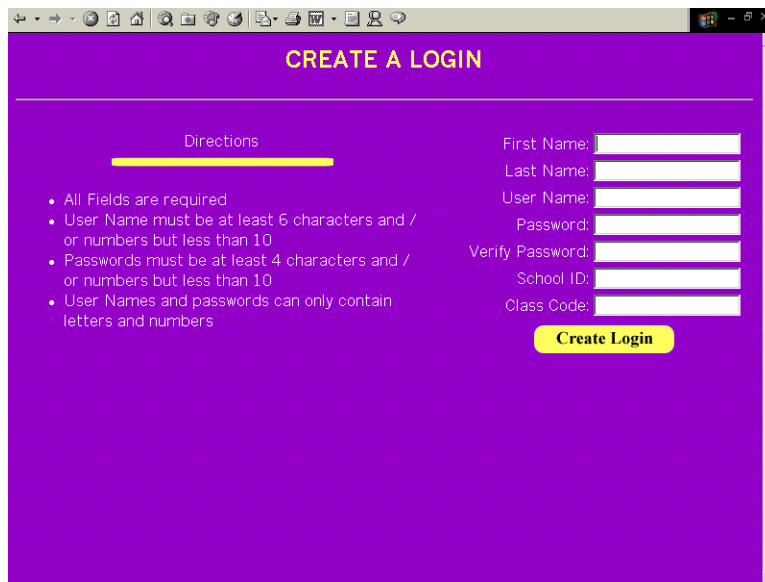
At the top of the login screen are the standard Metis logo, a button or link to the create a login page, and a standard login area consisting of login name, password and school. Initial or simple instructions on what to do on the screen are listed on the bottom half.



Figure 6. User login screen

2.4.3 Create Login Screen

The create login screen was added to reduce the amount of work that either the Metis administrator or the teacher would have to do. After the teachers account is setup, a classroom code will be given to the teacher to be passed on to the students that want to use this Web site. The students will enter the classroom code and be allowed to create their own logins and passwords. The logins adhere to a set of standards that are listed on the page. Examples of these standards are a password of at least four characters and a unique login name of at least six characters but less than ten.



The screenshot shows a web browser window with a purple background. The title of the page is "CREATE A LOGIN" in yellow text. Below the title, there is a section titled "Directions" with a yellow underline. The directions are listed as follows:

- All Fields are required
- User Name must be at least 6 characters and / or numbers but less than 10
- Passwords must be at least 4 characters and / or numbers but less than 10
- User Names and passwords can only contain letters and numbers

To the right of the directions, there are several input fields:

- First Name:
- Last Name:
- User Name:
- Password:
- Verify Password:
- School ID:
- Class Code:

At the bottom right of the form, there is a yellow button labeled "Create Login".

Figure 7. Create login screen

2.4.4 Failed Login Screen

The failed login screen displays a simply generic login failed message. It will not give the information that the login account does not exist or that the password is wrong; just that what they entered does not allow them into the site. After a three to five second wait, the page will automatically switch back to the main login page. A link will also be provided for browsers that do not support the automatic transfer.

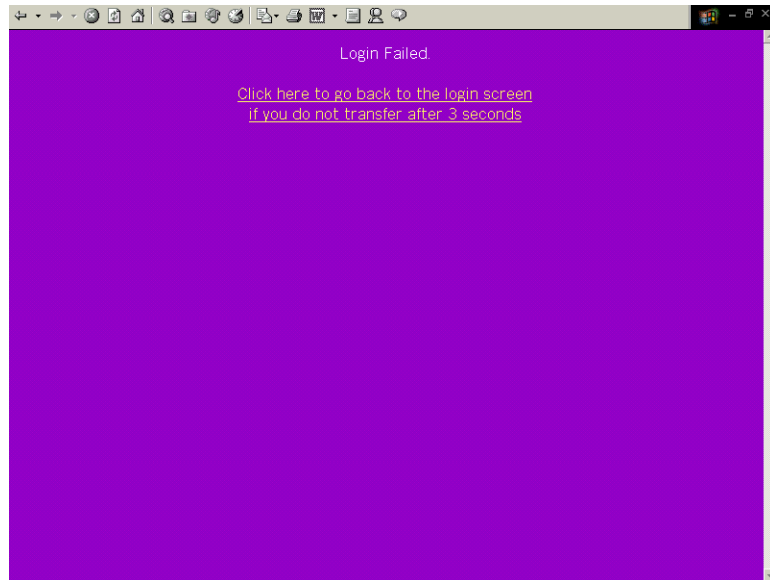


Figure 8. Failed or invalid login screen

2.4.5 Customized Menu Screen

The main menu has all the customization on it and each user could theoretically have a different screen. For example, if a student's class is only studying the American Presidents and mathematics; then those are all the flashcard links the student will see. There is no reason to allow this student the link for the Periodic table as this would be far too advanced for the student.

The teacher's screen will be the only screen that has the "Build a Classroom" link. Both students and teachers will have edit profile and report links.

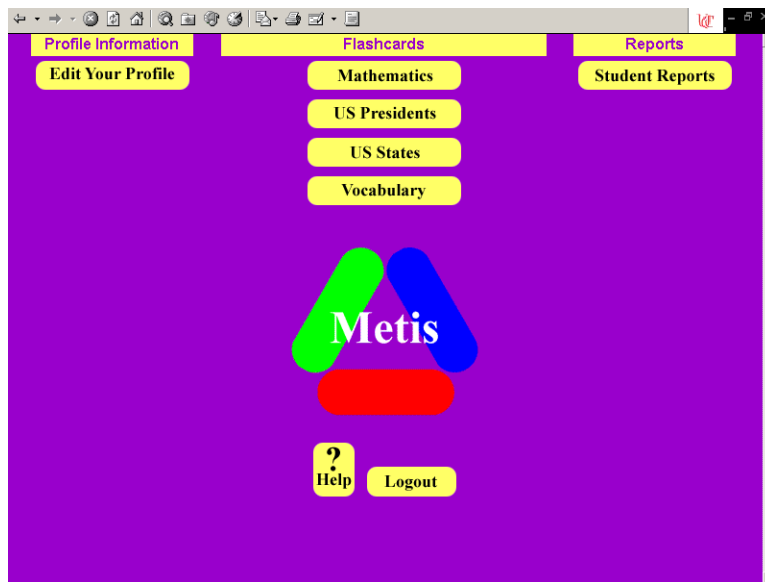


Figure 9. A student's customized menu screen

2.4.6 Build a Classroom

This screen allows each teacher to "build" their classroom. This is essentially collecting the data from the teacher that his / her student's will be studying. So a teacher could allow their students access to the US State flashcards and the mathematical flashcards. A second level of customization allows the teacher to select what portions of those flashcards to use. So for example, to only allow multiplication and division in the mathematical flashcards and skip the addition, subtraction, and rounding flashcards. A second example is to only allow the flashcards dealing with state capitols and abbreviations and not using the flashcards dealing with when states joined the union.

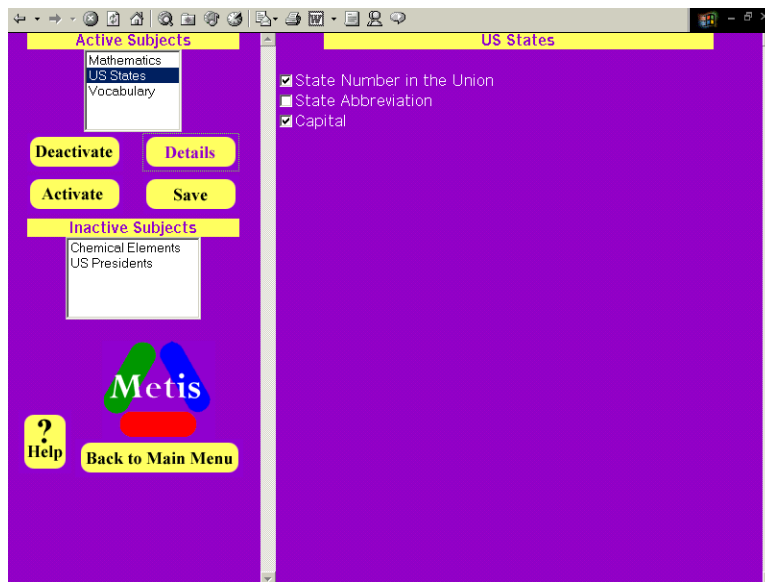


Figure 10. Build a classroom

2.4.7 Profile Information / Class Information

The students will be the only ones that have the access to add a classroom. One student may have two different teachers using this Web site and the teachers will have built two different classrooms but the student only has the ability to add one classroom when they create their login. This area will allow them to add another classroom to their login.

The screenshot shows a web browser window with a purple background. The title bar at the top reads "Profile Information". The page is divided into two main sections: "Profile Information" and "Class Information".

Profile Information

Edit Name Information

User Name:
First Name:
Last Name:

Change Password

Password:
Verify Password:

Update Name Information

Class Information

Classes Currently In

Drop Class

Join a Class

Class Code:

Join Class

Figure 11. Student's profile information / class information

2.4.8 Flashcards

This is the main part of the site. This is the actual flashcards used to interact with the students. The layout of the flashcards will be consistent across all content types. The flashcards will be on the right side and the menu will be on the left side. An example of the US States flashcards menu would be that the student has access to the state capitol and the state abbreviation flashcards. They may only want to work on state abbreviations, so they can just select that one type of flashcard. At the bottom of the menu will be a smaller version of the Metis logo that will take you back to the Customized Menu Screen. Also on the menu will be a link that will bring up a help document about how to use the flash cards.

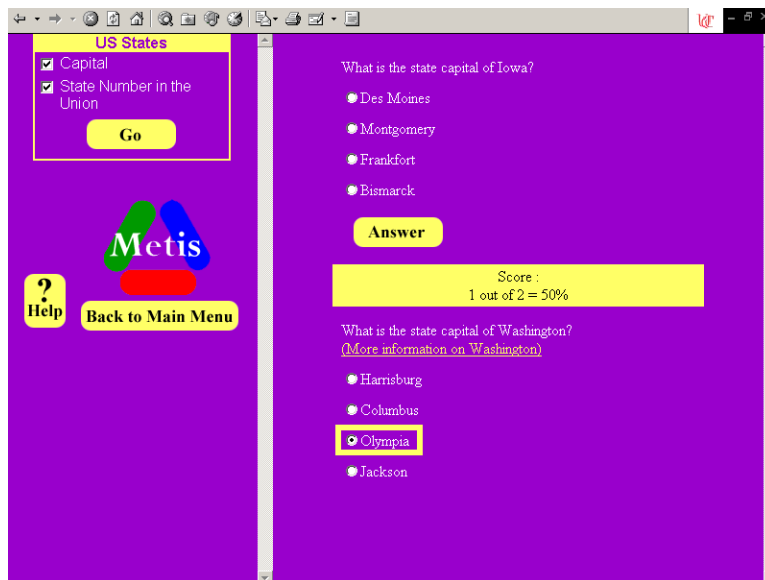
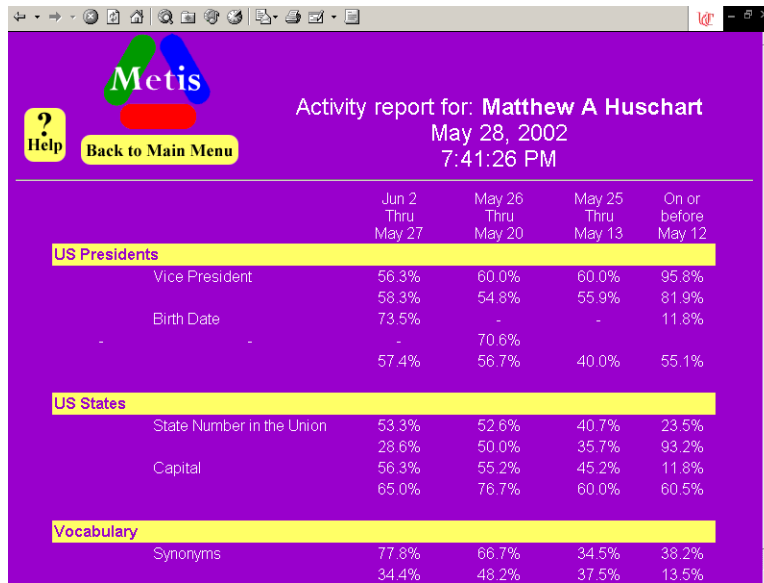


Figure 12. Example flashcard screen

2.4.9 Reports

Students and teachers will be able to run reports to check frequency of use and scores. Teachers will have reports showing their entire class and students will see their scores and the class averages or totals.



The screenshot shows a web browser window with a purple background. At the top left is the Metis logo. To its right, the text reads "Activity report for: Matthew A Huschart", "May 28, 2002", and "7:41:26 PM". Below the logo are "Help" and "Back to Main Menu" buttons. The main content is a table with four columns representing time periods: "Jun 2 Thru May 27", "May 26 Thru May 20", "May 25 Thru May 13", and "On or before May 12". The table is divided into three sections: "US Presidents", "US States", and "Vocabulary".

	Jun 2 Thru May 27	May 26 Thru May 20	May 25 Thru May 13	On or before May 12
US Presidents				
Vice President	56.3%	60.0%	60.0%	95.8%
Birth Date	58.3%	54.8%	55.9%	81.9%
	73.5%	-	-	11.8%
	-	70.6%		
	57.4%	56.7%	40.0%	55.1%
US States				
State Number in the Union	53.3%	52.6%	40.7%	23.5%
	28.6%	50.0%	35.7%	93.2%
Capital	56.3%	55.2%	45.2%	11.8%
	65.0%	76.7%	60.0%	60.5%
Vocabulary				
Synonyms	77.8%	66.7%	34.5%	38.2%
	34.4%	48.2%	37.5%	13.5%

Figure 13. Student report

2.4.10 Logout Page

The logout page will be a simply static page that thanks the user for using the Metis site.

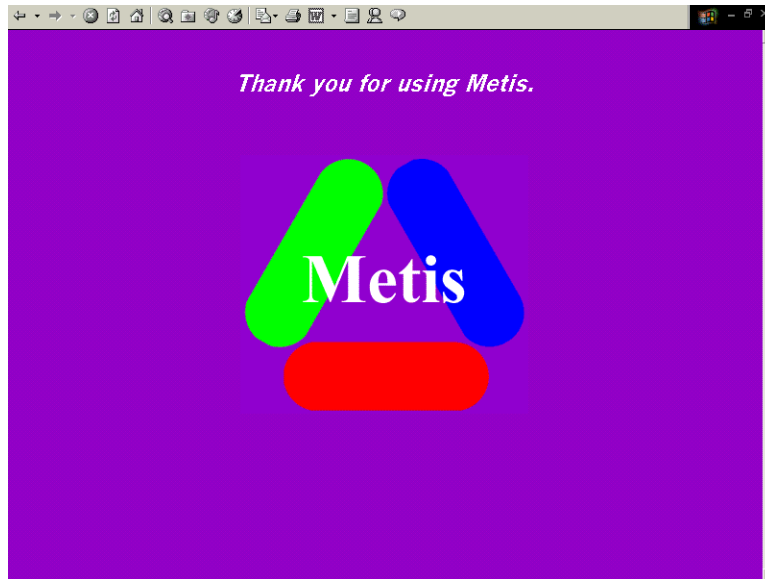


Figure 14. Logout page

2.5 Database Design

The database has been designed in a way that allows for closed-ended topics to be quickly imported into the database and used immediately. The database does not store the questions to present, it stores profiles, activity, answers, and possible wrong answers. The software will generate the questions.

The database conforms to the third rule of normalization with referential integrity being enforced.

Database relationships and table descriptions can be found in Appendix D.

2.6 Code

2.6.1 Visual Basic

The Visual Basic code generates almost all of the screens that the user sees. It randomly selects answers from the database, generates the question for those answers, and presents it with some wrong answers. It works this way so that the person adding new topics does not have to type in a long list of questions, just the topics that they would like flashcards presented on and links between the different topics for that subject.

Visual Basic was selected as the backend engine because it works well with many different databases, compiles to machine code for speed and security of code, and has good error handling capabilities.

The code will be written in one module without any forms. There is no traditional interface with the user because it is over the Web and the Main subroutine is set as the startup object. Three Window APIs (Application Programming Interface), GetStdHandle, WriteFile, and ReadFile, allow the application to communicate with the end user and accept information from a standard HTML post command.

2.6.2 Active Server Pages

Visual Basic Script will be used in the Active Server Pages and will be run on the server side. The ASP pages are in place to handle the security and initial creation of logins.

2.6.3 Java Script

Java Script will be used for all client side data manipulation and validation. This will include verification of the correct answers, updating current scores, formatting the questions in HTML, generating all the mathematical problems, and validating things like

password length. Java Script will also be used to react to the user's input. For example, when a user clicks the "Go" button in the flashcards, it will build the necessary URL string and submit it.

2.6.4 General

All the screens will contain standard HTML code. Many of the screens will use HTML frames to display the different information. A key to the frames is that there is normally one hidden frame that passes and accepts information to and from the Visual Basic executable. The Java Script then reads the information in this frame, reformats it, and presents it. The advantage here is that the screens that the user sees will appear faster and will not change as much because the communication to the Web site is happening and new visible pages are not being downloaded directly into the viewing area. A smaller amount of information also has to pass back and forth from the web site because the HTML formatting only gets downloaded one time and the current score is never downloaded.

2.7 Color Scheme

The color scheme will be consistent across all pages. The background of all pages will be a purple and the standard text on these pages will be in a white font. Highlighted text or text that is emphasized will be done in yellow. A yellow box will be drawn around closed-ended correct answers. Closed-ended flashcards are where the students pick the answer from a list of possible answers. Links or actions that will move between screen are done with graphics that change from yellow and black to yellow and purple when the user points at them. This was done to help the users with little

computing knowledge know that they can click there to make something happen. Each graphic also has corresponding text in the status bar to explain what it is going to do.

3. Objectives of the Project

1. A Web site that will aid in student's learning.
2. Light security access has been implemented.
3. The Web site was developed to allow teachers to select what subject matter flashcards their student will be able to access.
4. The students have the ability to create their own logins after they have been given a classroom code.
5. The Web site looks and acts very consistently so that the novice computer users will be able to learn and use it fairly quickly.
6. The initial site includes subject matter on US States, American Presidents, the Periodic Table, mathematics, and vocabulary.
7. The Web site incorporates all three learning stimuli: audible, reading, and writing (typing).
8. Reports were created that allow students and teachers to see activity and scores.
9. The Web site is compatible with Netscape Navigator 4.7 and above and Microsoft Internet Explorer 5.0 and above and the look of the site is very similar between the two competing browsers.

4. Proof of Design

Metis meets the goals outlined in this project. Metis is a Web based flashcard utility that is compatible across mainstream browsers and has a consistent look and feel so that inexperienced users can quickly learn how to use it. The flashcards on the Web site currently target the subjects of US States, US Presidents, Chemical Elements, Mathematics, and Vocabulary. All data with the exception of images and sound are stored in a Microsoft Access database that is accessed through either Active Server Pages

or the Visual Basic Metis executable. The database is setup so that new closed-ended topics or more vocabulary words can be added with minimal work.

Metis uses all three learning stimuli: visual, reading what is on the screen; audible, listening to the words the users have to spell; and touch, typing in the answers or selecting from a list.

Students have access to a report that shows their progress as compared to their classmates. Teachers have a report that summarizes the usage of their entire class with links that allow the teacher to drill down to a specific student's usage. All reports give a breakout of information over the past three weeks and then all other weeks.

A login, password, and class code are required for login and the URL in the address bar does not change from screen to screen. Implementing this light security on the Web site is to reduce the traffic and keep the Web site responding quickly to user requests.

Teachers select subjects from a list and then select topics for that subject that their students can study from or have access to. The students, in-turn, create their own logins and join the classes that their teachers have created. Both the students and teachers have control to update their names, user names, or passwords. These steps will help to reduce the time spent by the owner of the Web site administering to it.

5. Conclusions

Trying to make this application as portable as possible made coding it a little difficult since newer coding environments do not always create code that is compatible across platforms or Web browsers. I also could not make the application as fancy or

flashy as I would have liked. The Flash movie on the introductory screen may be the only thing that some users would not be able to use.

Working with the different frames was a challenge at the beginning because I was trying to hide a frame so that the user would not see the information in it. I could set the frame size to zero percent and this worked well with Microsoft Internet Explorer, but Netscape Navigator would still show a little sliver of information. I figured out that I needed to make the hidden frame last and assign all the space of the screen to the other frames. The last frame would then take the remaining space of the screen or nothing.

Here is some example code:

```
</FRAMESET>  
<FRAMESET COLS="30%,70%,*" FRAMEBORDER=0 BORDER=0>  
<FRAME SRC = "file1.htm" NAME=menu SCROLLING=Yes NORESIZE>  
<FRAME SRC = "file2.htm" SCROLLING=Yes NAME=problem>  
<FRAME SRC = "blank.htm" SCROLLING=No NAME=hidden NORESIZE>  
</FRAMESET>
```

I originally wanted to store everything in the database but came to realize that it was much easier to store the audio files in directories and store the relative paths to these files in the database.

I thought that collecting the data was going to be one of the easiest parts of this project. It turned out that the data was not always in formats or styles that make it easy to deal with (President Grover Cleveland's years in office and Vice Presidents for example). I also spent a good amount of time creating sensible wrong answers. For example, creating wrong state abbreviations for Ohio instead of letting the software generate or pick from the list of other states abbreviations. I did this because asking someone to select the state abbreviation for a state from a list of possible answers where only one

answer even contains letters from the state seemed to undermine the purpose of the Web site.

6. Recommendations

If I were to continue with this project, I would have to recommend an ActiveX component that allowed teachers to upload subject matter into the database. The ActiveX component should allow greater control and validation to where and what gets put into the database. I would not be as concerned that they could not use any Web browser to do this. Requiring Microsoft Internet Explorer 5.0 or greater to upload subject matter into the database should not prevent too many teachers from doing this. However, it might prevent teachers who do not use computers too often to not offer this to their students.

The program should ask questions in different manners. I started to put this code in (can see fields in the database for this in Appendix C) but I did not have enough time to finish it. The idea was that you do not always want to display the flashcard "What is the abbreviation of Ohio?" and give four possible state abbreviations; sometimes it should ask "OH is the abbreviation for what state?" and present the user with four states.

Even though the mathematics and vocabulary sections have the users type in answers, I would like to see an option for all questions to have text boxes for answer entry. I think after a student has gone through the flashcards a few times, it would be nice if they could turn on an option where they have to type in the answer instead of simply selecting it from a list.

I would add the capability of random questions and answers and not just flashcards on subjects that have 50 or so common data points. This would allow for flashcards on the War of 1812, or Christopher Columbus, or the American Revolution. This would require a question table that does not exist currently and the programming to randomly select a question and present one correct and four random answers.

I would also add a match up section. This would allow teachers to present maybe five items on one side of the screen and maybe twenty items on the other side of the screen and the students have to pick which items go with which.

Appendix A. Timeline

Item	Start Date	Finish Date
Draft Proposal	10/18/01	11/1/01
Progress Report 2	11/6/01	11/8/01
Revise Proposal and Develop Oral Presentation Materials	11/15/01	11/29/01
Design and Build Database	12/1/01	12/15/01
Collect Data for Questions	12/1/01	3/1/02
Implement Mathematical Part of Web Site	1/13/02	1/27/02
Implement Closed-ended Flashcards (user selects the answer from on the screen)	1/27/02	3/24/02
Implement Open Ended Flashcards (user types in answer)	3/17/02	5/1/02
Design Layout and General Look of the Web Site.	3/17/02	4/8/02
Create Graphics and Login Pages for Web Site	3/17/02	4/8/02
Work out all remain bugs from Prototype	5/8/02	5/15/02
Prepare Final Presentation and Report	4/25/02	5/19/02

Appendix B. Budget

Item	Source	Cost
Microsoft Office 2000 Professional	www.amazon.com	\$369.99
Adobe PhotoShop 6.0	www.amazon.com	\$564.99
Microsoft Visual basic 6.0 Professional	www.amazon.com	\$479.99
Resource Books and Materials <ul style="list-style-type: none"> • <i>Visual Basic 6 Interactive Course</i> • <i>Running Microsoft Access 2000</i> 	www.amazon.com	\$109.90
Web Hosting: US Net <ul style="list-style-type: none"> • 300 MB storage space • 30 E-mail accounts (<i>POP3</i>) • Unlimited e-mail aliases • Unlimited e-mail auto responders • 24/7 FTP access • Password protected directories • ASP (<i>Active Server Pages</i>) 	www.us.net	\$49.95 / month \$40.00 setup fee
Internet Connection: One.net	www.one.net	\$19.95 / month
Dell Dimension 4300 Computer and Monitor <ul style="list-style-type: none"> • Intel® Pentium® 4 processor at 1.6GHz • 128MB SDRAM • 40GB1 Ultra ATA/100 Hard Drive • 250MB Iomega Zip Drive • 17 in (16.0 in viewable,.26DP) M781s Monitor • 48x Max Variable CD ROM • Harmon Kardon HK395 Speakers • SB Live! Digital Sound card • Microsoft® Windows® ME 	www.dell.com	\$1,188.00
Total (one time fee)		\$3444.87
Total (monthly cost)		\$69.90

Appendix C. Database Relationships and Table Descriptions

1. Relationships

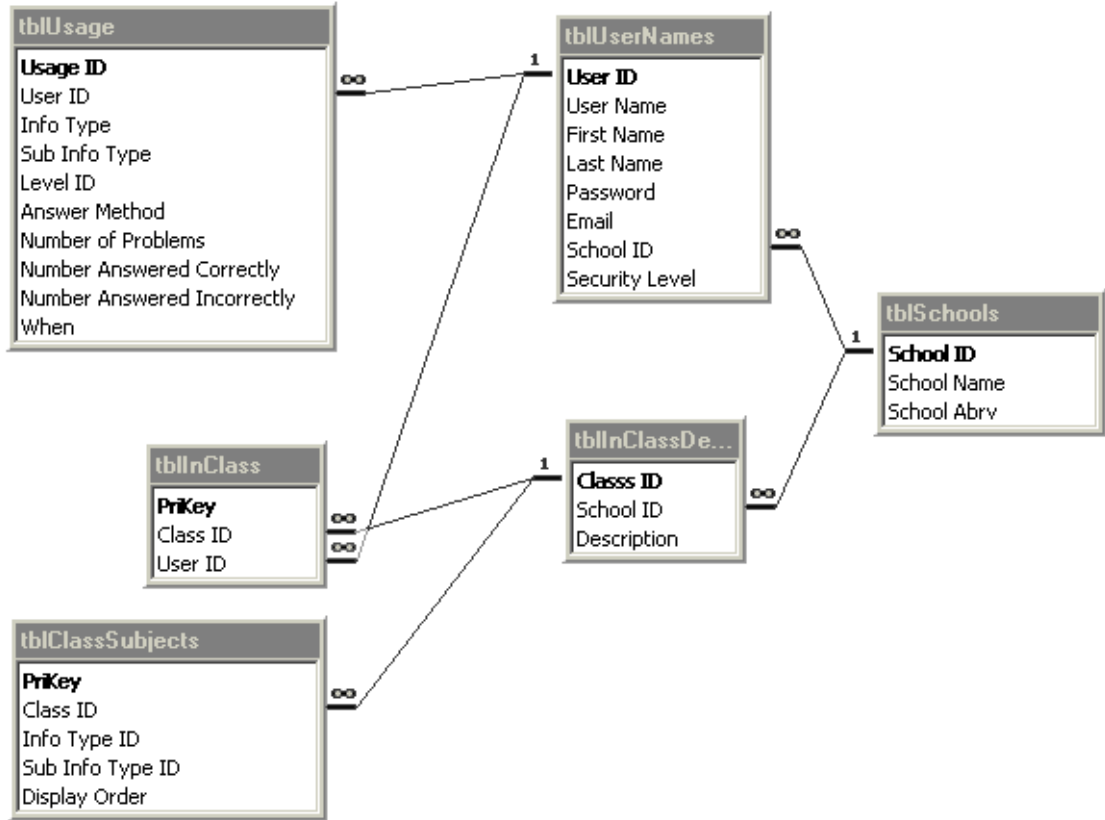


Figure 15. Database relationships (1)

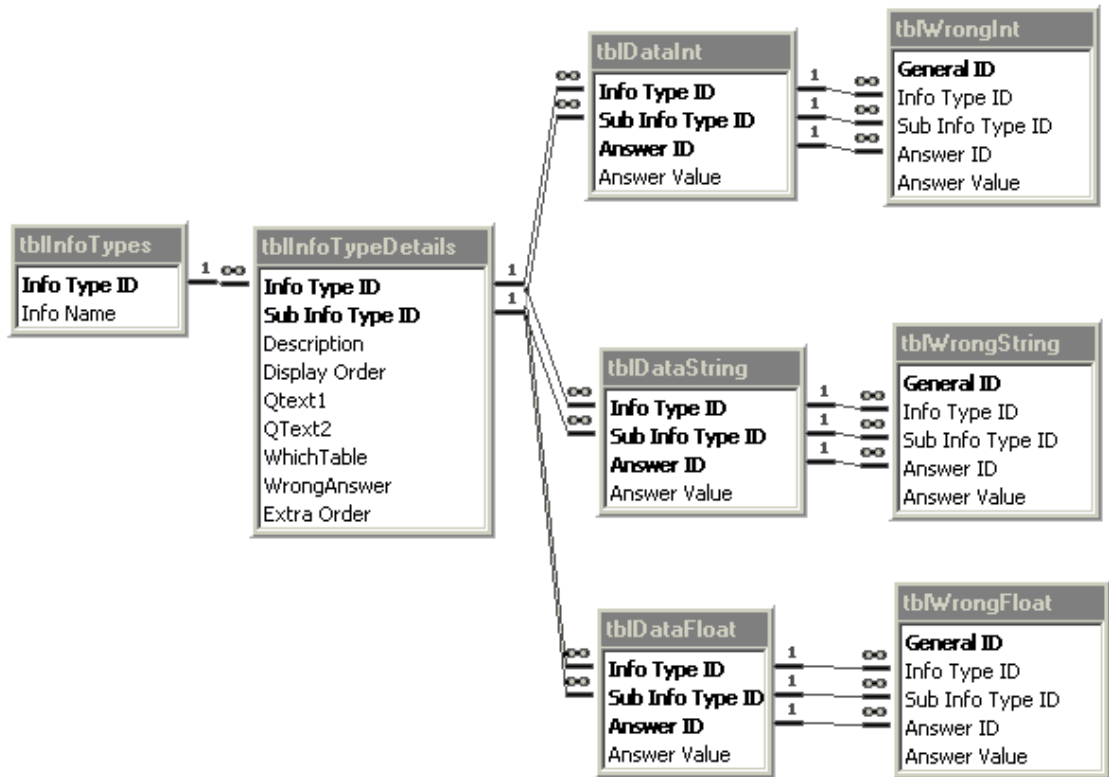


Figure 16. Database relationships (2)

2. Table Name and Descriptions

2.1 tblClassSubjects

Name	Type	Size
PriKey (PK)	Long Integer	4
Class ID	Long Integer	4
Info Type ID	Long Integer	4
Sub Info Type ID	Long Integer	4

TblClassSubjects holds what subjects or information types and sub-subjects each class is studying.

2.2 TblDataFloat

Name	Type	Size
Info Type ID (PK)	Long Integer	4
Sub Info Type ID (PK)	Long Integer	4
Answer ID (PK)	Long Integer	4
Answer Value	Double	8

TblDataFloat is the table that holds all floating point value answers. These are things like boiling and melting points of chemical elements.

2.3 tblDataInt

Name	Type	Size
Info Type ID (PK)	Long Integer	4
Sub Info Type ID (PK)	Long Integer	4
Answer ID (PK)	Long Integer	4
Answer Value	Long Integer	4

TblDataInt is the table that holds the correct answers for all questions whose answers are integers or whole number. It is designed to allow for numbers in the range of 2,147,483,648 to -2,147,483,647.

2.4 tblDataString

Name	Type	Size
Info Type ID (PK)	Long Integer	4
Sub Info Type ID (PK)	Long Integer	4
Answer ID (PK)	Long Integer	4
Answer Value	Text	255

TblDataString is the table that holds all the string or text answers. For example, a state's capital city name.

2.5 tblInClass

Name	Type	Size
PriKey (PK)	Long Integer	4
Class ID	Long Integer	4
User ID	Long Integer	4

TblInClass holds what classes each student attends.

2.6 tblInClassDetails

Name	Type	Size
Class ID (PK)	Long Integer	4
School ID	Long Integer	4
Description	Text	50
Class Code	Text	50
User ID	Long Integer	4

TblInClassDetails stores the descriptions of each class and the unique class code that is required to join a class. The User ID field in this table is not the user identification number for the current user, but rather the user identification number for the teacher of the class.

2.7 tblInfoTypeDetails

Name	Type	Size
Info Type ID (PK)	Long Integer	4
Sub Info Type ID (PK)	Long Integer	4
Description	Text	50
Display Order	Integer	2
Qtext1	Text	255
QText2	Text	50
WhichTable	Text	50
WrongAnswer	Byte	1
Extra Order	Long Integer	4

TblInfoTypeDetails stores the information that describes each set of flashcards. This information ranges from how to ask the questions, to what questions can be asked, to where the information is stored in the database. WrongAnswer describes where wrong

answers can be pulled from. A value of 1 will only pull wrong answers from the answers for other questions with the same sub-informational grouping. A value of 2 will pull answers from where both the wrong answer tables and the correct answer tables. Finally, a value of 3 will only pull answers from the wrong answers tables.

2.8 tblInfoTypes

Name	Type	Size
Info Type ID (PK)	Long Integer	4
Info Name	Text	50

TblInfoTypes stores the main headings or names for the different flashcards. For example, “US States” or “Periodic Table”.

2.9 tblMathLevels

Name	Type	Size
Class ID (PK)	Long Integer	4
AllowedLevel	Integer	2

TblMathLevels holds what levels of complexity are offered to the user.

2.10 tblSchools

Name	Type	Size
School ID (PK)	Long Integer	4
School Name	Text	100
School Abrv	Text	10

TblSchools stores the information about the individual schools that are using Metis.

2.11 tblUsage

Name	Type	Size
Usage ID (PK)	Long Integer	4
User ID (PK)	Long Integer	4
Info Type (PK)	Long Integer	4
Sub Info Type (PK)	Long Integer	4
Number Answered Correctly	Long Integer	4
Number Answered Incorrectly	Long Integer	4
When	Date/Time	8

TblUsage tracks how the students are doing and is used to drive the reports for Metis.

2.12 tblUsageDetails

Name	Type	Size
Usage ID (PK)	Long Integer	4
User ID	Long Integer	4
Info Type	Long Integer	4
Answer ID	Long Integer	4
Block	Yes/No	1

TblUsageDetails is used to keep track of which flashcards the current user has had recently. When all the flashcards in a particular set have been asked, the block field is cleared and all the flashcards can be presented again.

2.13 tblUserNames

Name	Type	Size
User ID (PK)	Long Integer	4
User Name	Text	50
First name	Text	20
Last Name	Text	50
Password	Text	10
School ID	Long Integer	4
Teacher	Yes/No	1

TblUserNames stores all the logins for Metis. This table also keeps the information as to who is and who is not a teacher. This information directly links to what

screens and reports the user has access to.

2.14 tblVocabClasses

Name	Type	Size
PriKey (PK)	Long Integer	4
ClassID	Long Integer	4
VocabID	Long Integer	4
Chapter	Long Integer	4

TblVocabClasses holds what makes up each individual chapter for each class.

2.15 tblVocabSynAnt

Name	Type	Size
SynAntID (PK)	Long Integer	4
VocabID	Integer	2
Type	Long Integer	4
Word	Text	50

TblVocabSynAnt holds the synonyms and antonyms that are used in the vocabulary section. A value of 1 for the type field represents synonyms and a 2 value represents antonyms.

2.16 tblVocabWords

Name	Type	Size
VocabID (PK)	Long Integer	4
VocabWord	Text	50
NewWord	Boolean	1

TblVocabWords holds the words that will be used for the vocabulary section.

The actual sound files are held in a "vocab" subdirectory and all files are simply named by the word that it is used for. The NewWord field is set to true when a word is added to the database but it does not exist as a .wav file.

2.17 tblWrongFloat

Name	Type	Size
General ID (PK)	Long Integer	4
Info Type ID	Long Integer	4
Sub Info Type ID	Long Integer	4
Answer ID	Long Integer	4
Answer Value	Double	8

TblWrongFloat holds wrong floating point answers for different questions.

2.18 tblWrongInt

Name	Type	Size
General ID (PK)	Long Integer	4
Info Type ID	Long Integer	4
Sub Info Type ID	Long Integer	4
Answer ID	Long Integer	4
Answer Value	Text	255

TblWrongString holds wrong integer answers for different questions.

2.19 tblWrongString

Name	Type	Size
General ID (PK)	Long Integer	4
Info Type ID	Long Integer	4
Sub Info Type ID	Long Integer	4
Answer ID	Long Integer	4
Answer Value	Text	255

TblWrongString holds wrong text answers for different questions.

Appendix D. Proposed Testing

Obviously as each part of the site will be tested as it is programmed and at the end of programming, the entire site will be re-tested on multiple browsers. Here is a list of specific area that will be tested.

1. Welcome Screen

- Does the Flash movie play?
- Does the link take the user to the login page correctly?

2. Login Screen

- Are valid logins accepted?
- Do the following scenarios work correctly? (C = Correct, I = Incorrect or Blank)

Login	Password	School	Pass / Fail
C	C	C	
C	C	I	
C	I	C	
C	I	I	
I	C	C	
I	C	I	
I	I	I	

3. Create Login Screen

- Are blank required fields (login name, password, school, classroom code and name) rejected with appropriate response?
- Is a login name of less than six characters or more than ten accepted?
- Is a password of less than four characters accepted?
- Is an invalid school id rejected?
- Is an invalid classroom code rejected?
- Is a valid classroom code for a different school rejected?
- Are valid entries accepted and a login created?
- After a successful login creation, are you returned back to the login page?

4. Failed Login Screen

- Does this transfer you back to the login page after three seconds?
- Does the link transfer you back to the login page for those browsers that do not support the automatic transfer?

5. Customized Menus

- Is the correct content listed for each classroom?

- Does each link take you to the correct flashcards?
- Do the students have the ability to add a classroom?
- The students should not have the ability to create a classroom, do they?
- Do teachers have the ability to build a classroom?
- The teachers should not have the ability to add a classroom, do they?

6. Build a classroom

- Are all available subjects listed?
- Are all assigned subjects listed?
- When you select a subject from the list, do the question types about that subject show correctly?
- Are changes made on the screen saved correctly; both adding and removing data?
- Does the link take the user back to the Customized menu screen?

7. Add a classroom

- If a classroom code is blank, is a correct message displayed?
- Does an invalid classroom code get rejected with the correct message?
- Does a valid classroom code for a different school than the current student get rejected with the correct message?
- Does a valid classroom code get accepted?
- Does the link take the user back to the Customized menu screen?

8. Flashcards

Mathematics

- Do only the operators that the user is allowed to use show up and are the others hidden?
- Do the levels show up correctly?
- Does the help link work?
- Do the addition problems have the right answers?
- Do the subtraction problems have the right answers?
- Do the multiplication problems have the right answers?
- Do the division problems have the right answers?
- Does the remainder selection box work correctly?
- Do the rounding problems have the right answers?

Closed-ended answers

- Are the 4 answer choices for a question randomized (does the correct answer move around)?
- When a question comes up a second time, has more than 90% of the flashcards been answered?
- When a question comes up a second time, are the answers in a different order and possible different answers altogether?
- Does it grade the student's input correctly?
- Does it respond correctly when a student does not select an answer?

Open Ends

- Can the user type in the answer?
- Does a blank answer give the correct response?
- Does it grade the student's input correctly?

Spelling

- Are the words played correctly?
- Does it grade the student's input correctly?

9. Report Screen

- Do the reports run and produce output?
- Do the reports for students only show that student's information and class aggregate?
- Do the teacher's reports show information for all the student's that have accounts?

10. Logout Page

- Does it display properly?

Notes

- A. The Medical College Admission Test (MCAT) is a standardized, multiple-choice examination designed to help admission committees in selecting which of their applicants will be accepted into medical school.
- B. The Top 100 is a Web site dedicated to rating today's top sites. All the ratings and reviews are submitted by users.
- C. Ivan Pavlov, 1849 - 1936, was a Russian Scientist who experimented with learned behavior psychology. He would ring a bell each time he feed food to his dogs. After a while, he would ring a bell and the dogs would start to salivate because they thought they were going to get food.
- D. HTML stands for HyperText Markup Language and this is the language for the surfing the Internet. Files may end (the file's extension, after the period) in "htm" or "html".
- E. American Online (AOL) is one if the most popular browsers in use today for the simple fact that they are one of the largest ISPs (Internet Service Provider) around. Even though AOL and Netscape have become one company, AOL still uses a form of Microsoft's Internet Explorer as their browser.
- F. ASP stands for Active Sever Page and is different than the standard static HTML page. ASP pages have the ability to have logic or code embedded into them and react or display different information depending on different circumstances.
- G. CGI stands for Common Gateway Interface and is a standard for interfacing external applications with Web servers. A plain HTML document that is delivered to a user is static, which means the information exists in a constant state: a plain text file that doesn't change. A CGI program, on the other hand, is executed in response to a user's request so that it can create dynamic pages.
- H. The mathematical page will be downloaded each time a new math session is started. When the user leaves the site or changes areas of information, for example going to the American President area, the information to drive the mathematical page will be lost and will need to be downloaded again on next entry to the mathematical area.
- I. The name for the project is Metis. Metis was the Greek Goddess of wisdom and knowledge. This site is about the search for knowledge, wisdom, and good grades.

References

1. Armstrong, Alison and Charles Casement. *The Child and the Machine: How Computers Put Our Children's Education at Risk*. Beltsville, MD: Robins Lane Press, 2000.
2. Associated Press. "Schoolwork and the Internet: Americans see blessing and pitfalls". <http://www.msnbc.com>. August 28, 2001.
3. DiSessa, Andrea A. *Changing Minds: Computers, Learning, and Literacy*. Cambridge, MA: 2000.
4. Ferguson, Penny. Teacher. C.O. Harrison Elementary. Personal Interview. October 8, 2001.
5. Howard, Jeffrey. "Principles of Good Practice in Service-Learning Pedagogy". <http://www.uu.edu/centers/faculty/resources/article.cfm?ArticleID=45>. November 25, 2001.
6. Idocs Inc. "Idocs Guide to HTML". <http://www.idocs.com>.
7. Lahey, Benjamin B. *Psychology: An Introduction*. Dubuque, IA: 1989.
8. Magro, Jennie. Special Education. Oak Hills High School. Personal Interview. August 30, 2001.
9. Sharp, David. Science Department Head. Elder High School. Personal Interview. October 8, 2001.
10. Viescas, John. *Running Microsoft Access 2000*. Redmond, WA: Microsoft Press, 1999.
11. Web 100, The. "Education and Reference". <http://www.web100.com/listings/education.html>. November 25, 2001.
12. Webb, Martin. Internet Related Technologies. <http://www.irt.org>.