Requirements Specification
for
eFlash

Final Version
Revision 1.2

Authors: David Chu, Sean Gabriel, Daniel Honegger, Darren Lam,
Ying Tat Ng, Aretha Samuel, Kenneth Wong, Tony Wu, Anthony Yee

Date: February 15, 2006

PROJECT ABSTRACT

The eFlash team is designing a study-skills application designed to eliminate a student's need for paper flash cards. Our software intends to replace these vital learning tools while improving on the issues inherent in a pen-and-paper system.

DISTRIBUTION NOTES:

Review Deadline Date: _____
Review Meeting Date: _____

Comment(s):

_____
_____
_____
_____

Reviewer List:

Distribution List:

Document Revision History

Revision	Date	Author(s)	Comments
0.1	2006-02-10	Gabriel	Initial draft, rough division of labor
0.2	2006-02-12	Wu	Customer profiles, methodology
0.3	2006-02-12	Lam, Ng	Networking overview (tentative)
0.4	2006-02-13	Honegger	Competitive landscape, security, etc.
0.5	2006-02-13	Yee	Market overview and analysis
0.6	2006-02-14	Chu, Gabriel, Wong, Samuel	Overview and requirements compilation, graphics added
0.7	2006-02-15	Gabriel	Working version with revisions
1.0	2006-02-17	Gabriel, et al	Finalized with networking component
1.1	2006-02-23	Honegger	SWOT analysis
1.2	2006-04-10	Gabriel	All issues resolved

Product Overview

Purpose

This specification describes requirements for the eFlash learning software in subsequent sections. While providing an overview of the requirements for this tool, the document covers the market and customer base intended for this application. Customers are free to reference this specification throughout the design process, and we have made every attempt to ensure that customer needs have been realized in the following sections.

Short Description

In our first team meeting, our marketing expert realized that she had a distinct need for automated self-testing when she was enrolled in a foreign language class. Upon further investigation, we determined that a market existed for this particular application of learning software, and thus eFlash was born. Our primary goal is to use technology to improve upon all of the shortcomings of traditional paper flash cards, with the hope that customers will find our product to be superior enough to discontinue their use of paper flash cards in the future.

To this end, we have ambitious plans to feature something for everyone, retaining the versatility that makes paper flash cards a worthwhile solution in the first place. Though flash card software has been developed in the past, its scope and comprehensiveness have left something to be desired, and this is where we hope to step in with eFlash, a compelling, unobtrusive study-skills application.

Authors

Although marketing has provided the meat of this document in terms of translating customer needs into software requirements, all team members have played a vital role in translating this document into its current state today.

Name	Team Role
David Chu	Development
Sean Gabriel	Project Leader
Daniel Honegger	Architecture
Darren Lam	Backstop
Ying Tat Ng	Development
Aretha Samuel	Marketing
Ken Wong	Visionary
Tony Wu	Development
Anthony Yee	Development

Target Audience

This document serves to inform customers of our plans for eFlash, as well as developers who will begin to create the program as it is laid out in the following sections. The requirements section remains the heart of this document, and both parties should consider it in equal regard to ensure that it meets the needs of the customers while being quantized appropriately for the developers.

Specification Flow

This specification is the first document written after the eFlash project was approved for development. It depends mainly on customers, marketing, and development input before its conception into actual software.

Dependencies

Customer input will be received into this requirements specification and adapted for inclusion into requirements as appropriate. Such changes will reflect the evolution of our product as we continue to receive more and more customer feedback.

Review Considerations

There are two distinct review processes for this document which practically guarantees that there will be at least two passes of this document. Customers need to review the document for completeness and to ensure the customer needs are documented correctly, while developers and the core team need to ensure there is enough information presented in this document to write a functional specification. Before this document is finished, the core team must feel comfortable that it is possible to execute the baseline deliverables as outlined.

Requirements Overview

Introduction to requirements

Students from all academic disciplines need to quiz themselves constantly over the topics they learn in their classes. Flash Cards have been proven to be extremely helpful for students' self-study habits. However, the current procedure of manually creating paper flash cards can be exceedingly unmanageable and time-consuming, and can therefore be radically improved with the proper software application. Students currently spend lots of money and time either manually making paper flash cards or buying pre-made ones. They then have to categorize the cards and confirm that they are not misplaced. These actions can be tedious and inefficient. Additionally, these cards can sometimes become transparent enough so that the answers on the back of the cards become visible to the students. This unfortunate feature limits the students' ability to self-test their knowledge. Secondly, music and art students are generally at a disadvantage when it comes to quizzing themselves over the material they learnt in their classes. These students are not able to use flash cards to quiz themselves over the different art and music pieces they have seen and heard. Therefore, a suitable software application must be created that will aim to resolve all of these problems.

eFlash is the answer because it can efficiently store, and categorize all different types of student-created flash cards. This software allows for many different types of input, such as text, sound, and picture. The user can record his/her voice, and input those sound recordings as descriptions or themes. The user can also upload any images or sound-files saved on his/her computer to the eFlash database, which will then be used by the application to create customized multiple-choice and matching quizzes. The application can also simulate the flash-card learning process by utilizing the computer screen as a flash card. All these features are offered so that students may use eFlash for as many different purposes as possible. As a bonus, eFlash will also enable users to post their flash cards publicly online so that many different students can share flash cards. Students can download these publicly-shared flash cards online and can modify them on their own computer. This feature will save students even more time, as they do not have to create these flash cards themselves.

References / Contacts / Terminology

References:

N/A

Contacts:

Summer Chrisman
Senior, UC Berkeley

Rocio Ramirez

Junior, UC Berkeley

Kurt Keutzer
Professor, UC Berkeley

Terminology:

Flash card – A card printed with words or numbers that are briefly displayed as part of a learning drill.

Limitations

N/A

Requirements

Functional Requirements

I. Inputs

Baseline Deliverables

- The user must be able to input text, sound, and picture into the database along with its subsequent descriptions.
- Each word could potentially have these features: Description/Definition/Translation1, Translation2, input from user, Sentence using the word, Tenses, Category, Picture, Sound file. The user need not provide this much information, but he or she can use as much as is needed. For example, a user might want to manually record the correct pronunciation for the word or upload a sound file for a particular word. See Figure 1.
- Have color options when inputting words and definitions; therefore the user is able to distinguish between words and definitions.
- As the user is inputting his/her words into the database, the program must search the database to see if the word is already written. The program will accomplish this through the "find-as-you-type" method.
- In addition to manually inserting pictures and descriptions, as well as terms and their definitions/translations (see Figure 3), the user should also be able to supply a formatted text file that will display information about many words and definitions. Our program should be able to pick out all the words and definitions from this format and create flashcards from this information. The output can also then be exported into a PowerPoint presentation format (see Figure 2).
- For audio/music input, our program will extract answers from ID3 tags of the mp3 files and automatically create music flash cards. (see Figure 4)

Ideal Solution

- For audio/music input, our program will produce a PowerPoint presentation with embedded music samples.
- For language-learning users, the program will be to take in more than 2 translations, possibly 3 or 4. This will help the users study more than just 1 foreign language.

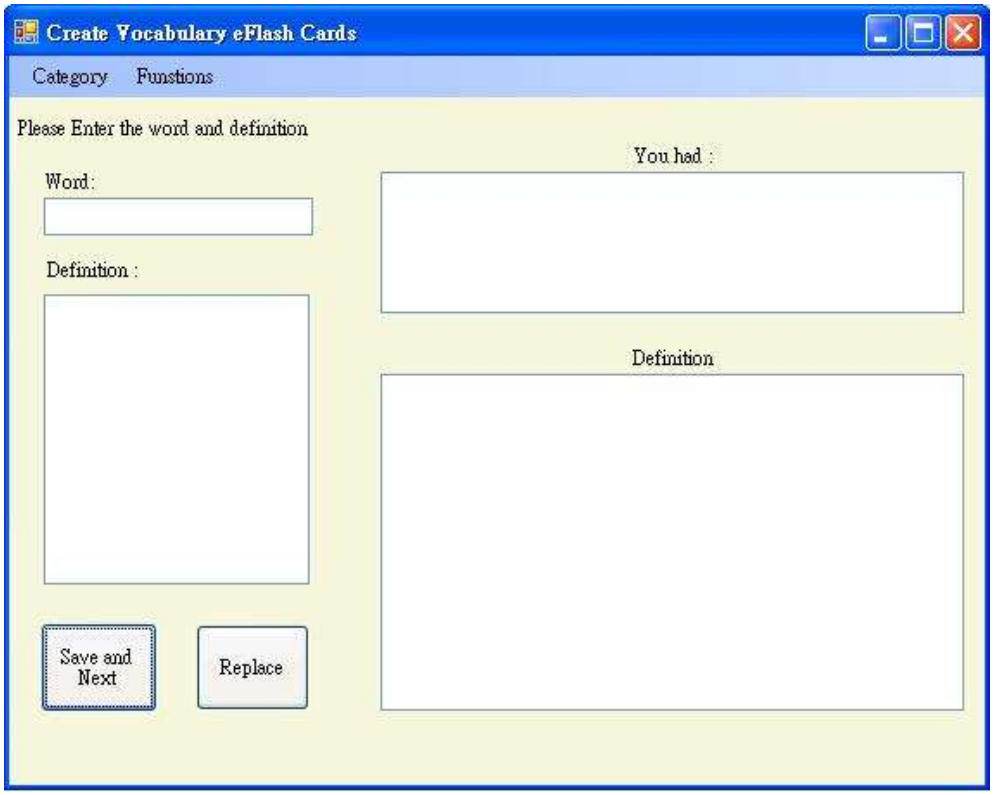


Figure 1. Interface for adding words to the database.

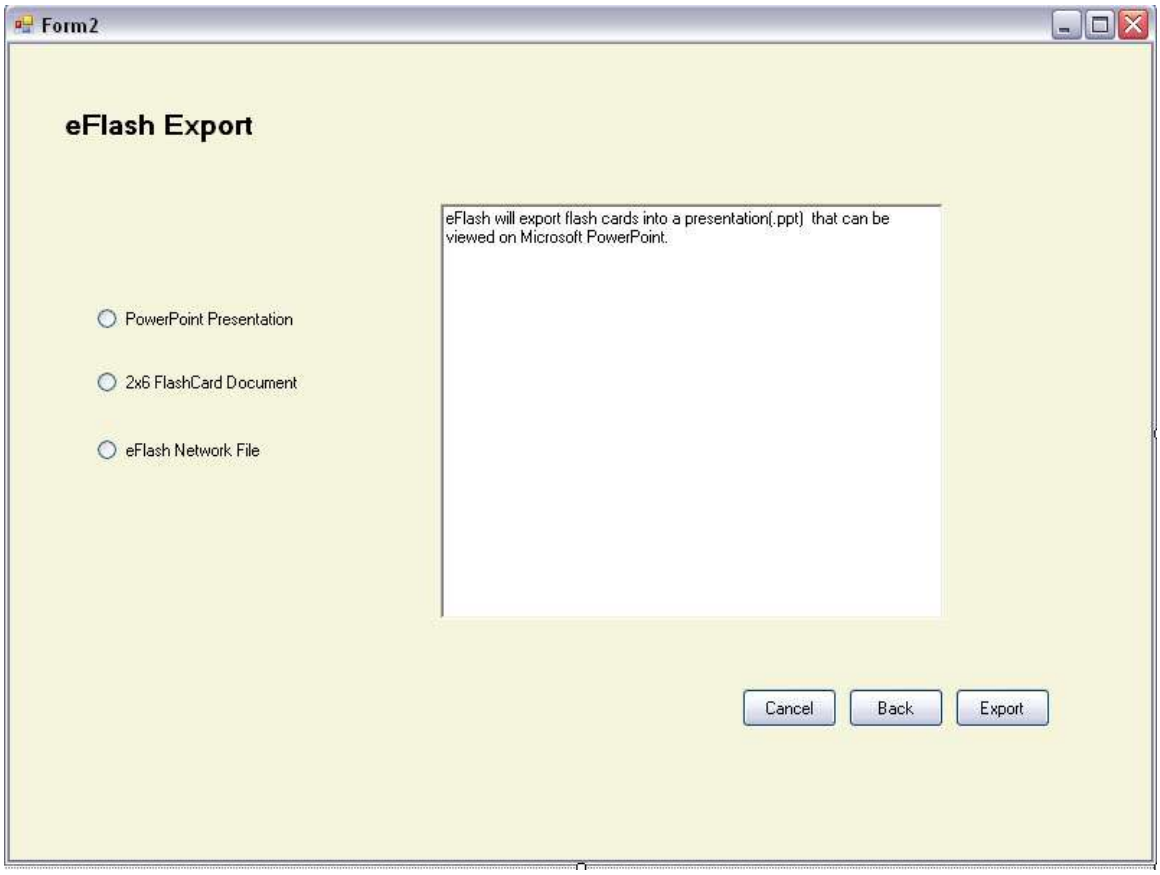


Figure 2. Exporting flash cards to a desired format.

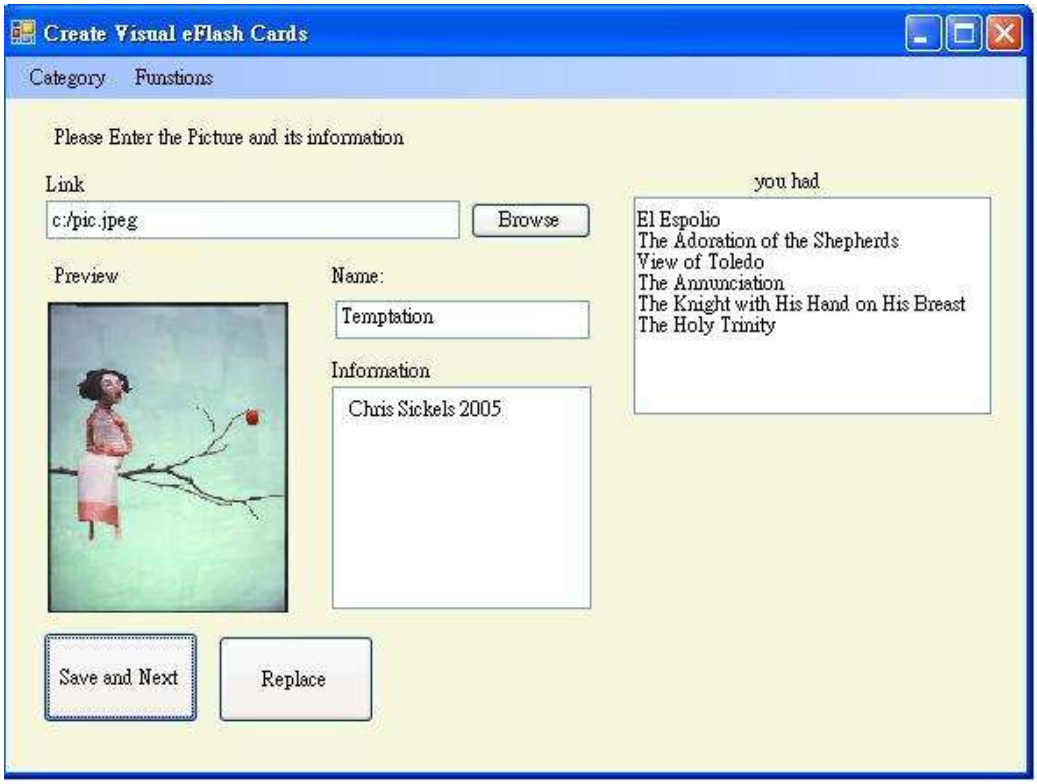


Figure 3. Users may input images and their descriptions.

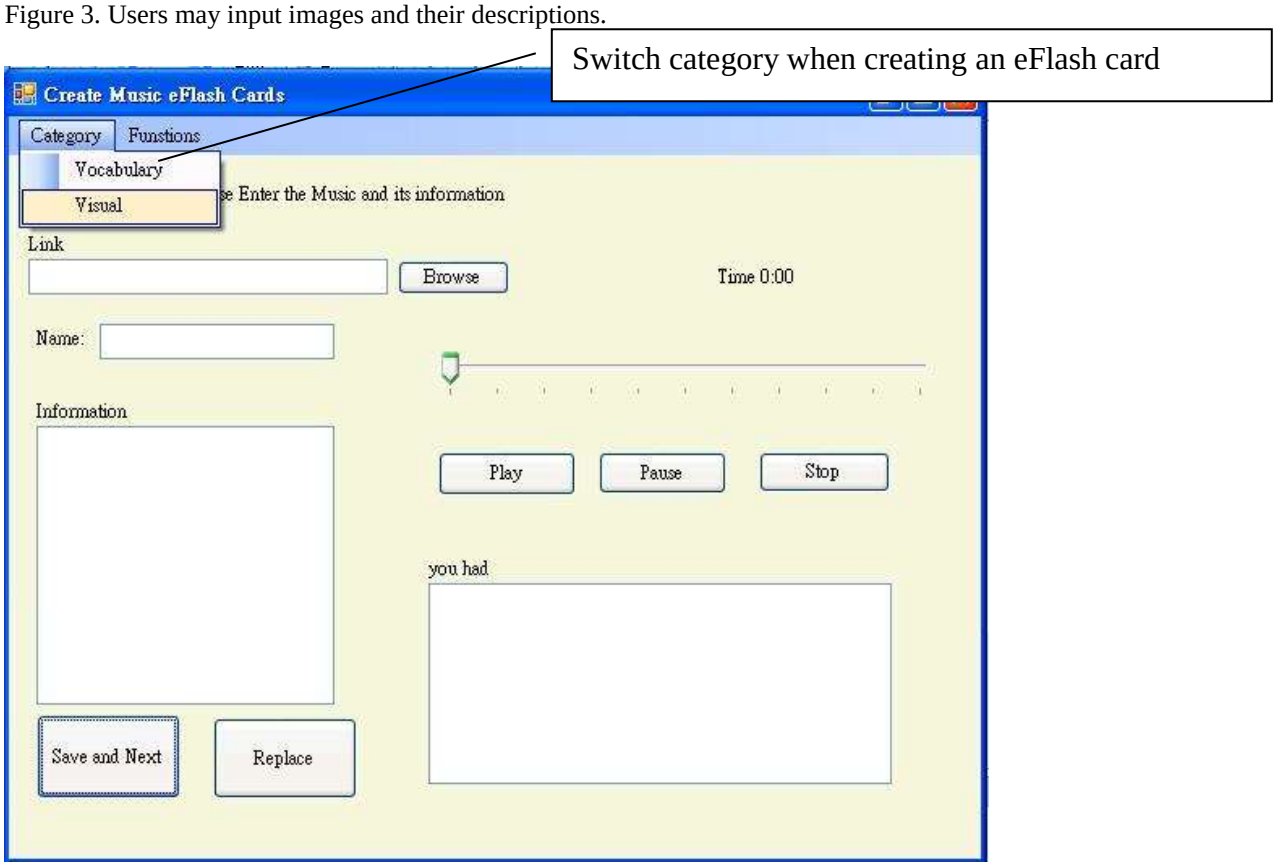


Figure 4. Creation of Music/Audio Flash cards.

II. Quizzing Options

Baseline Deliverables

Display

- The program will display simulated flash cards of a particular user-specified category. (see Figure 5).
- In addition to flashing the word/picture on the screen, the program will also play the sound file associated with the word.
- The word/picture should be displayed in the middle of the screen until the user is ready for the next word. Timed quizzes are also in the realm of possibility.
- The user should also be able to move through the flash cards by pressing the keys instead of clicking with the mouse.
- If the user has 2 different translations, each being in a different language, then the flash card will flash both translations.
- Some users would like to test themselves without opening our eFlash program. They want a set of flash cards to be ordered as PowerPoint slides, so that these slides may be viewed outside the eFlash program. These slides will take on the order of word/picture --> definition/description, with the word/picture being on the first slide and its description/definition being on the consequent slide.

Quizzes

- The program will also provide matching quizzes, with words on one column and their definitions on the other.
- The program will automatically create multiple choice quizzes based on the user's choice of what category he/she would like to be tested in.

Ideal Solution

- To target biology students' needs, eFlash could also include the option of having the users upload their images and using the program to annotate them. For example, a biology student will upload human body diagrams of the heart or skeleton, and using the program, they can mark up the picture with their own notes and descriptions. Essentially, the user will have a picture from biology, click a point in it, write something associated with a point, and when he or she drags the mouse over it, the user will see a pop-up for the definition of the point.
- To target history students' needs, eFlash could have the option of allowing the user to create customizable timelines.

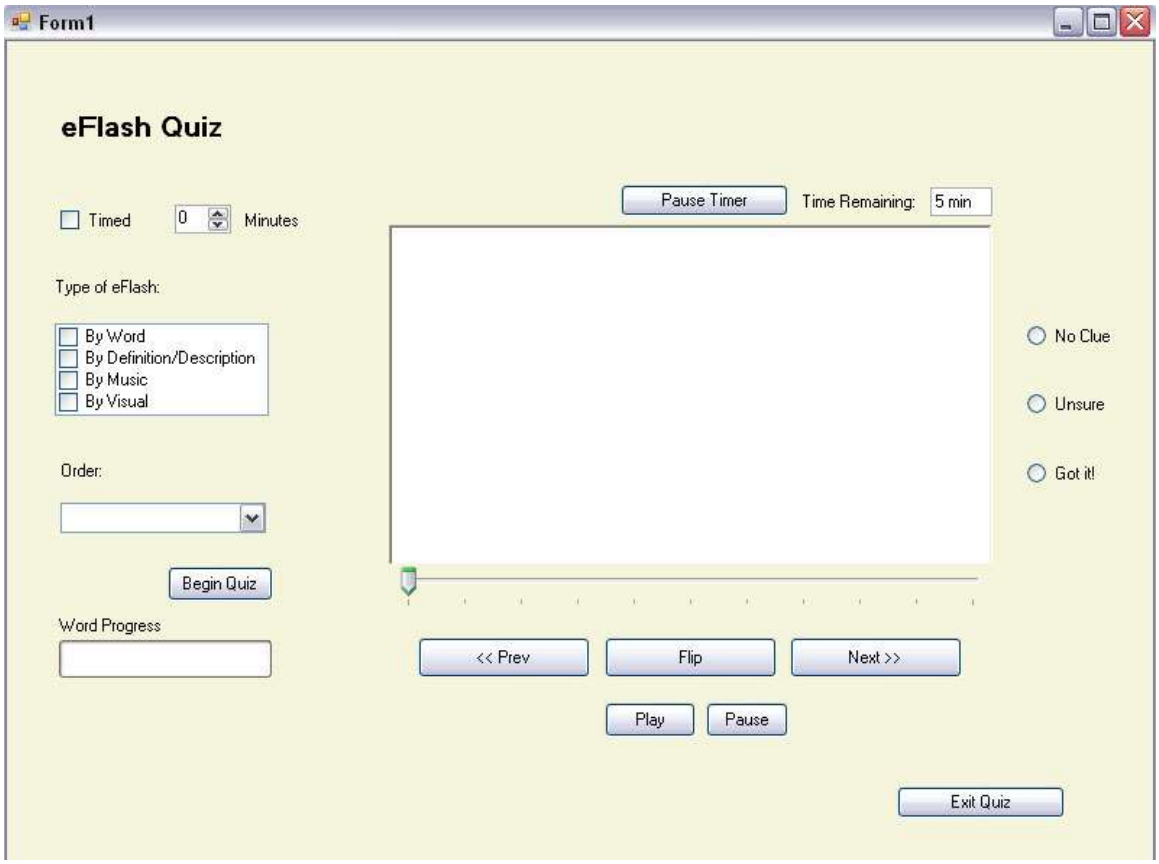


Figure 5. Quiz Testing.

System Requirements

- Operating Systems Supported: Windows XP
- CPU Speed: 300 MHz or above
- Memory: 256 MB
- Free Disk Space Required: 100 MB
- Connection: 56k minimum (Required only for eFlash Card Sharing)

User Interface Requirements

Baseline Deliverables

- For Flash Card Showing, the user prefers that the word should be plainly placed in the middle of the screen with no fancy graphics. The simpler the flash card, the better.
- The user would also like peruse through the slides by using the keyboard instead of using the mouse.
- Users may also want to color-code their inputs and definitions according to their own tastes, and we will allow them to do so if that is their wish.
- Users would like a clear, presentable, easy-to-use main menu where they can easily access older flash-card projects or create new ones. (See Figures 6 & 7).

Ideal Solution

- Users may want the flexibility to sample a portion of the music file for their music tests. Instead of the whole file, the user can now choose the particular section of music that is most efficient for remembering songs.

Figure 6. A nice clean interface for the main menu.

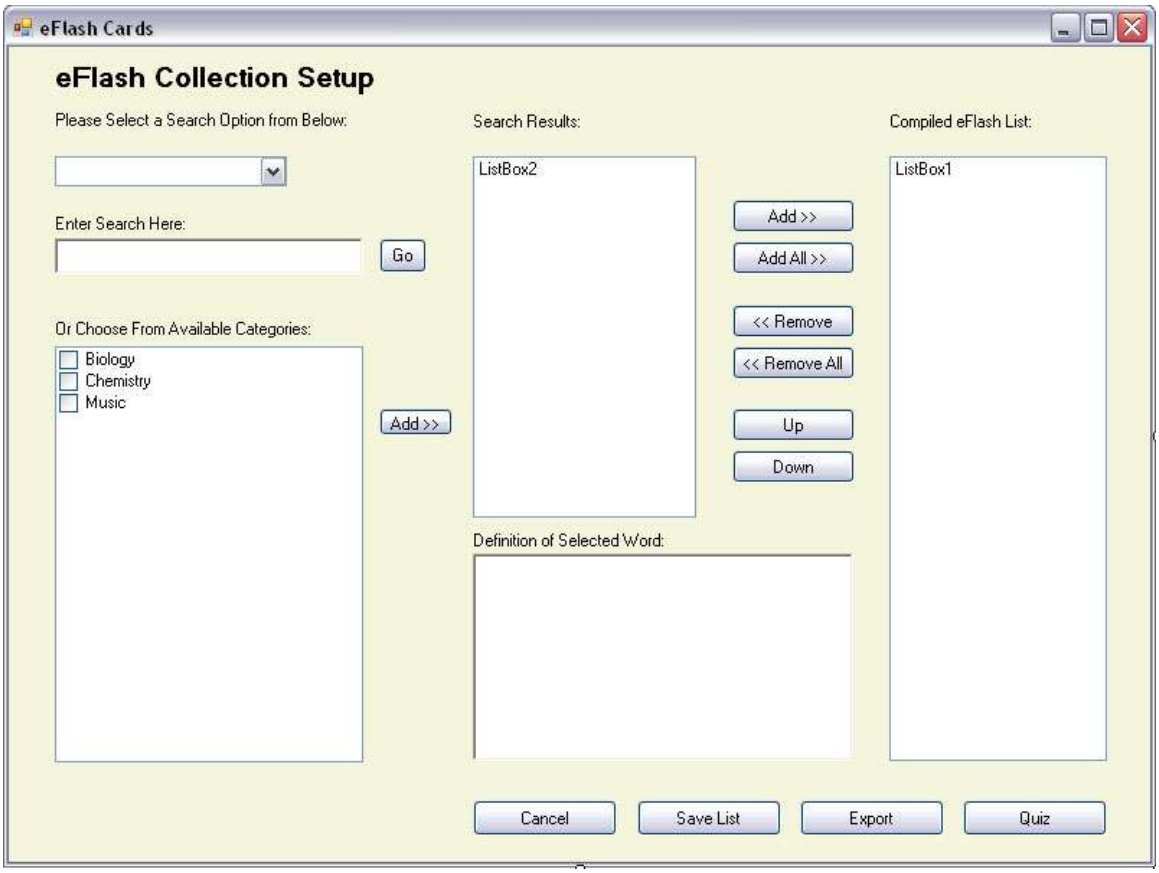


Figure 7. A representation of how a user may select different flash card projects.

Other Interface Requirements

Network Requirements

Users may connect to a centralized database server which provides convenient Internet-based file sharing and management services (see Figure 8). Once a user creates an account ~~on the server~~, the program would provide an interface which allows users to upload or download their decks of flashcards to/from the server by means of SQL queries (see Figure 9). The centralized database server collects and maintains a list of user submitted flashcard documents organized into categories which facilitates flashcard sharing among users.

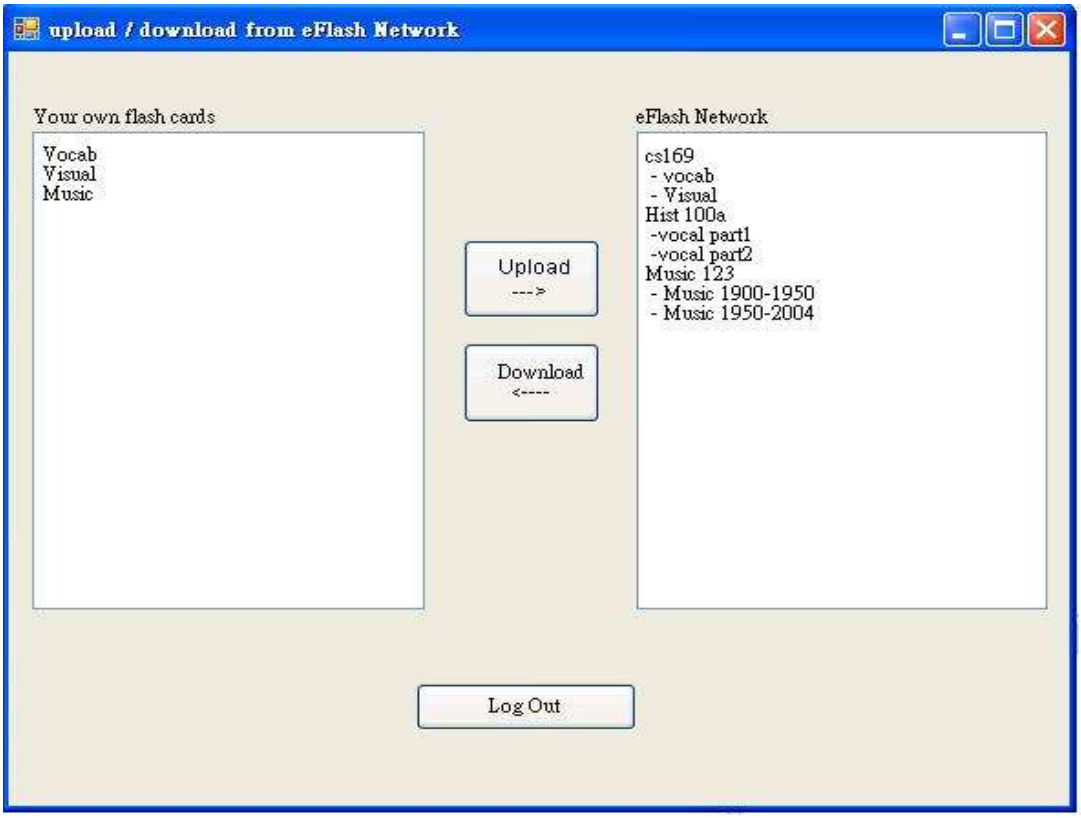
Portability

- Users' flashcard documents on the server can be downloaded at anytime, anywhere in units of deck with our program installed. The client program behaves like a thin query generator.
- Users are allowed to synchronize the files in their computers and their files on the central server.

Scalability

- Users can share/un-share their files with other users by deleting decks off the network that they previously uploaded.
- A user can view and download other users' files that are open to public by browsing the categories in the public view box.
- Once a user has selected the files to be download, the files can be transferred to the user account by pressing the "Download" button.

(Recently Revamped)
Figure 8. eFlash Network logon UI.



(Recently Revamped)

Figure 9. Simplified interface for uploading and downloading from the network.

Security Requirements

Under the local model: not applicable. As our program is a locally run application operating on mundane, publicly available data, we assume no security precautions need be built into our software. If multiple users operate the software on the same machine, we assume they are trusted users and can freely create, modify, and delete any user-profiles.

Under the networked model: Identifying information will be kept with each uploaded deck, such that other users can search for decks created by a specific login. Also, only authenticated login's can alter their previous uploads by either removing them from the repository or updating a deck. The networking component of our client will not be enabled until the user chooses to use it in such a fashion, so we intend to make them aware of the benefits/risks of the online option at such a time.

Internationalization Requirements

We have not as yet found or discussed searching for any non-English speaking customers, so all menu options and instructions will be in English. However, as one of the primary functions of eFlash is as a foreign language study aid, a large range of characters must be supported. The Unicode / ISO 10646 standard[1] provides by far the widest universal character support, and will be used for all flash card information storage.

[1] Information is available at http://www.unicode.org/ and http://www.cl.cam.ac.uk/~mgk25/unicode.html

Market Overview

Market Analysis -- General Market Overview

Overall market review

Despite a recent slump in computer hardware, the computer software industry remains a strong component of the global economy, especially with the adoption of computers as a standard platform for doing business. Because it is so easily accessible to newcomers, the industry is intensely competitive and fast paced. As a result, the industry is heavily invested in the research and development of new and better products. Companies are continuously pressured to deliver better technology every year. Delays often translate into substantial losses and even bankruptcy.

Among other markets in its industry, the education and training software industry shows promise for substantial growth because of the increasing need for trained professionals in industry. Like other markets in the industry, the education software market is easily accessible to newcomers. Major players often secure exclusive deals forcing smaller companies to merge with large ones. With the recent convergence of computers and other electronic devices, the market will intersect other markets such as cell phones and PDA's.

A growing trend among education software providers is the migration to Internet based applications. Education software providers are developing new technologies that expand services to Internet, creating a new class of applications referred to as Web services. The shift to web based applications could dramatically change the distribution model of software eliminating the need for packaged software entirely.

Revenue potential

Total Computer Software Sales

Year	Retail Volume (billion U.S. $)
2002	301.8
2003	289.7
2004	330.4
2005	383.3

The industry has been characterized by explosive growth following the growth of computer hardware that continues to grow approximately 15% each year, generating annual revenues of approximately $300 billion (US). However, there are signs that the industry is reaching maturity. While some markets still show promise for substantial growth, many software markets in the industry have stalled. As a result, while larger companies remain profitable, smaller specialized companies are competing in a revenue pool that has recently stopped expanding quickly enough to support them all. Such factors are likely to lead to massive consolidation within the industry. Perhaps a counter-intuitive business trend in the industry is the growing number of open source applications becoming available. These

applications are provided free of charge and generate revenue from licensing code and associated training and services.

Market characteristics, opportunities, and driving forces

- **The education software market is a relatively new industry and has not realized its full potential for growth.** Commercial computers have only been available for 20 years. Since then, the market has boomed into a $32 billion (US) industry. The rapid industrialization of foreign countries such as China and India promises even more growth.

- **Migration to computer services combined with a larger demand for education is creating new opportunities for education software.** The rising standard of education is creating a growing demand for professionals to train them. Education software provides a cost effective solution for providing a standard level of education for a large number of people. Schools are increasingly integrating interactive learning software into their curriculum. As a result, the education software industry has a large consumer base that will continue to grow.

- **Relatively cheap production and distribution costs make the education software industry easily accessible to newcomers.** New companies are continually entering the education software market providing competitive alternatives to existing products. As a result, companies in the market are pressured to deliver newer and better products by certain deadlines in order to compete with others in the industry. Delays and buggy software often lead to substantial losses.

- **The competitive nature of the market necessitates constant investment in new technology.** The industry is fast paced and driven by the research and development of better software. Companies must continuously look for ways to improve their technology so as not to be outdone by competitors.

Market Analysis -- Competitive Landscape

Competitors' positions

A large slew of free flash card programs exist and are easily downloaded from the web. A list of the most prominent of these competitors with their strengths and weaknesses is presented in tabular form on the next page (Table 1). To complete the SWOT analysis, simply take the competitor's weakness as possible opportunities for eFlash to surpass the competition, and competitor strengths as threats to eFlash's market share.

A general opportunity in the digital flash card market is increasing market share through high exposure. There is essentially no traditional advertisement competition; digital flash card resources are seen as utilities that are sought out or not, so a high position on internet search engines, coupled with an intuitive, easy-to-use design is the best way to gain users.

Product	*Strengths*	*Weaknesses*

WinFlash Basic[2]	online deck support, user statistics, multi-media support	clumsy UI, too many poorly organized options
Flash Card Exchange[3]	online-based repository, many quiz modes / games, printable, downloadable in many formats, import cards from excel / notepad	no sound support, personalized statistics not available, some online user created decks very poor quality
CueCard[4]	very basic, easy to use, international menu options	text/image only, manual entry, severely limited option
Memory Lifter[5]	established test methodology, multi-media (sound / image / text), test scheduler, many entry options	too many entry fields for most applications, manual input
FlashCard Wizard[6]	clean interface, 3-sided card, customizable, set filtering, set grouping / merging, statistics	manual input, text only, bland testing presentation

Table 1. Tabular presentation of competitor's strengths and weaknesses, functionality-wise

Competitive differentiators and barriers

Our generic functionality will duplicate many available and supported programs. Our only competitive edge in the area of general purpose flash-cards is a cleaner, more attractive GUI component. Where our product goes beyond the competition is in its wide array of uses, from sound-card abilities for those studying music, to less supported languages such as Tibetan. Our comprehensive program will also offer a large array of outputs: either quizzing the user inside our program, outputting the card set in a slideshow format (PowerPoint), or printing them out. Additionally, we plan to implement optional heuristics for focusing on areas of higher difficulty.

Benchmarks

First and foremost are the obvious functionality benchmarks: does our program eventually perform the task we say it will, namely allow the creation, storage, use, and sharing of digital flash-cards? Each type of media flash-card must be tested independently. Additionally, response time must be critically tested, specifically in areas involving large media types. The response time should be essentially instantaneous, or have only the slightest load delay (less than one second) for large decks or decks constructed with high-resolution images or sound-files. Finally, online uploading and storage of cards must be tested under a variety of connections. Tests where the uploader time-outs are necessary to ensure that the system properly deals with partial uploads: only display as available for download fully uploaded decks.

Metrics

[2] http://www.openwindow.com/pages/compare.htm
[3] http://www.flashcardexchange.com/
[4] http://www.wadeb.com/cuecard/
[5] http://www.memorylifter.com/
[6] http://www.foolsworkshop.com/fw.html

We will evaluate the implementation of our software on a group of beta-testers comprised of our close customers. Specific areas for evaluation are the intuitiveness and cleanliness of the GUI, speed of program response (require instantaneous), appropriate output format, and overall usefulness as a tool to streamline the creation, organization, and use of digitized flash cards. As our primary goal is to remove the painful parts of flash-card use, we must make sure to meet this goal; the software must reduce user-time spent making cards and effectively organize and present specific testing regimes.

Support

The purpose of eFlash is to provide an intuitive stream-lined alternative to the analog task of flash card creation and maintenance. As such, the GUI will be as simple and self-explanatory as possible. Helpful features will include pre-built card set templates for specific uses (i.e. a foreign language template). Step by step instructions for creating decks will also be displayed online at the eFlash™ website.

Licensing

Given that our product's primary use is to stream-line an already widely practiced analog experience, our software will be distributed freely under a public license agreement allowing dissemination and use by anyone. The source code (and hence any further modifications and updates) will be reserved solely for eFlash™.

Customer Profiles -- General Profile

The customers of eFlash will consist mainly of students of various levels, studying material that requires some form of memorization. Some examples include foreign language students memorizing words and phrases, high school students building a vocabulary list in preparation for SAT exams, biology students studying anatomy and molecular structures, history students learning key historical events, etc. Many of these students already employ the use of physical flashcards to help with their memorization process, but are frustrated with the annoyances of dealing with these paper cards, such as difficulties in managing large stacks of cards, card randomization, and so on. Ideally, they would like these processes to be automated.

In addition, our product will also be valuable to teachers of the subjects mentioned above. They recognize the usefulness of flashcards, but have no feasible method of producing and distributing physical flashcards to their students, and so would appreciate the network flashcard-distribution system of eFlash.

Because our target customers include many students and teachers in humanities and science fields, we expect them to have little to no computer literacy, and so the user interface must be kept as simple and intuitive as possible while retaining functionality.

Out of the customers we interviewed, some favored simplicity, while others wanted more features and functions. Our product will need to be tailored to suit both of them.

Customer Profiles – Example Profiles

Name: Summer Chrisman
Profession: UC Berkeley Undergraduate Student
Date of Interview: 2-9-06
Work environment: Chrisman primarily uses flashcards to aid her study of Swahili, but also finds them useful for biology. She feels that they are unwieldy to manage and organize, and sees digital flashcards as a possible solution to her problems.
Feedback: Chrisman emphasized that the software should be quick and simple to use. All data for each flashcard should be input on the same screen and navigation should be done with keys instead of mouse clicks for greater efficiency. The cards should be kept simple, with one font in black and white. The card-creation process should be made hassle-free by the use of templates. During flashcard review, the window should be maximized to full screen to minimize distractions. It would also be nice to be able to import audio files and play them automatically when each flashcard is viewed. Additionally, she wants a quiz function that uses multiple choices and/or mix-and-match pairs. She also expressed the desire for an advanced search function, allowing her to search through the database of flashcards with a search-as-you-type feature. Finally, in addition to Swahili, she could also use our product for biology and history. For these subjects, the flashcards need to support the importing of images and/or embedded links. These images should also be annotatable: the user can click on any point on the image to add a text caption/annotation. During quizzes, these captions can then be tested in a fill-in-the-blank manner.

Name: Rocio Ramirez
Profession: UC Berkeley Undergraduate Student
Date of Interview: 2-10-06
Work environment: Ramirez initially began using physical flashcards when she started learning Portuguese. Now she finds them useful for other languages as well.
Feedback: Ramirez wanted a variety of various features in flashcard format and management. First, it is important to be able to easily customize the overall layout of each card, specifically controlling the color of the text. For instance, she wants to be able to specify to have two columns on one flashcard, with each column having a different color. Additionally, Ramirez wanted to be able to create flashcards with more than two sides (used to put multiple languages on a single card). Second, she suggested a grouping feature. Cards are grouped into categories, specified by the user during creation, and all tenses of the same word are grouped onto the same card. When reviewing and quizzing, she can then choose which categories to use. Finally, like Chrisman, Ramirez also emphasized the need for sound support, preferably being able to record her own voice with our program. These are then either played during reviews/quizzes, and/or used to correct her own intonation.

Name: Kurt Keutzer
Profession: UC Berkeley Computer Science Professor
Date of Interview: 2-8-06
Work environment: Keutzer has been continuingly learning various alphabets over the years. He will mainly use our product to learn Tibetan script, but also various works of music. It is important that the flashcards be extremely portable and easy to use.

Feedback: Keutzer prefers to view the flashcards independently from our software. Thus, high up on his list of priorities is that our product be able to easily (with one click) export the flashcards to a PowerPoint (PPT) presentation file, in which the slides alternate between questions and their corresponding answers. If possible, it would be best if the slides appear in a random order. Because this will be his main viewing method, he will not require any quizzing mechanisms. As far as input goes, Keutzer will supply our product with a list of sequentially named JPEGs or other image files (containing the Tibetan script) and a corresponding list of answers for each image. Our software should then automatically read in these files and generate the appropriate flashcards. Similarly, for music flashcards, he will provide us with a list of MP3 files. Instead of a list of answers, however, eFlash will extract the music answer information straight from the files' ID3 tags. The user can then choose which fields they wish to be tested on. Additionally, for each music flashcard, Keutzer would like to be able to highlight a specific section of the music, which can then be tested on a "sound byte." Finally, he wants to be able to provide our software with an outlined list of data, out of which we will generate the flashcards. It would be ideal if eFlash can read in Word documents, but Wordpad (.rtf) files are also acceptable.

Effect on Methodology

eFlash will revolutionize the use of flashcards by eliminating the annoyances and limitations of physical flashcards while providing a wealth of additional features. Nearly all of the time-consuming organization and management processes will be automated, allowing the user to spend more time on the actual studying. Additionally, the functionality of flashcards is vastly enhanced, expanding upon current flashcard technology as well as extending out to fields never before possible (such as music). With eFlash, students across countless fields will be able to enjoy the benefits of flashcards without the associated headaches.

Dependencies / Assumptions

We have no formal dependencies for our product outside of the minimum system requirements listed earlier in this document. In particular, export features such as PowerPoint slide deck creation are intended to be modularized, such that a user without a PPT installation can still make use of this export functionality.

Issues

Resolved

- We will actively pursue the addition of network functionality to eFlash, after confirming customer needs for such a feature.

Checklist

The following checklist is provided to help the reviewers (and author) prepare for the review by providing a set of questions for the reviewer to answer about the specification document. If the answer to any question is "no", that item should be identified as an issue at the review. The checklist is only a guideline; it should not be solely relied upon for a complete review. Reviewers may want to add their own questions to the checklist.

Y	N	CONTENT
___	___	Do the requirements state the customers needs?
___	___	Do the requirements avoid specifiying a solution?
___	___	Do the requirements avoid specifying a design?
___	___	Are you satisfied with all parts of the document?
___	___	Do you believe all parts are possible to implement?
___	___	Is each part of the document in agreement with all other parts?
		COMPLETENESS
___	___	If the requirements properly prioritized?
___	___	If there are requirements imposed by third party products, are those requirements listed?
___	___	Are all of the necessary interfaces specified, including input and output?
___	___	Are all performance requirements (e.g. speed, memory, capacity) specified?
___	___	Where information isn't available before review, are the areas of incompleteness specified?
___	___	Are all sections from the document template included? If not, is there reason for the changes?
___	___	Are you willing to accept the specification with the items in the open issues section unresolved?
		CLARITY
___	___	Are all requirements reasonable?
___	___	Is the level of detail of each requirement appropriate?
___	___	Are the requirements written in a language appropriate to the audience?
___	___	Are all items clear and not ambiguous? (Minor document readability issues should be handled off-line, not in the review, e.g. spelling, grammar, and organization).

Appendix: Methodologies Employed to Create the Requirements Specification

User requirements were gathered from three semi-structured interviews. Below are the original notes from each interview and a summary of its main use cases.

Interview 1:
Interviewee name: Kurt Keutzer
Notes:

Tibetan script

- Continuing to learn alphabets over the years, when you think you've learned them all then there's always one more that needs to be studied
- Hard to scan in a giant sheet and just recognize what needs to be taken in…
- Instead, he will work with our format
  - Supply a list of sequentially named jpegs or other image files
  - Supply a list of sequentially named "answers" for each jpeg
  - Our program should **export to a PPT presentation**, create a slide deck
    - (Why? So the flash cards are "platform independent")
    - (Note: could leverage OpenOffice code here)
  - Standardized output allows us to be interoperable
  - One-click export would be nice, or just match up file to answers
- No scoring/statistical features necessary, will be used to "self-test"
- Make a slide deck like: question, answer, question, answer, etc.

Music study

- Same idea here, but instead of a list of jpegs, provide a list of mp3s
- Better idea: let's **extract answers from the ID3 tag** of the mp3 files
  - User can select which options they want to be quizzed on
  - Sound files can be embedded in PPT to the best of my knowledge
    - (If not, XML/web app is an acceptable second)
  - **Randomization** of the generated output would be nice, is there a way to randomize the order of a PPT presentation on the fly?
- When sounds are embedded, would be nice to replay the section if available
- Great UI idea: **highlight** chorus/refrain of the mp3 to be tested on a "sound byte"

Use cases:

- The user has a number of images of Tibetan alphabet, each with a corresponding definition/answer, and wishes to create a deck of digital flashcards in PPT format. He opens the eFlash software and inputs two lists: one for the images and another for the list of answers. He then clicks a button in the eFlash GUI, upon which eFlash creates a PPT presentation consisting of slides that alternate between the given images and their corresponding answers. This PPT file is put into a user-designated directory for the user to retrieve.
- The user has a number of MP3 files, each with certain information in its ID3 tag. He runs eFlash and inputs a list of the files. eFlash then displays in its GUI a list of the imported sound files. The user goes through the list, highlighting a specific section of

some files using some form of visual display (a waveform, slider bar, etc.). He then clicks a button in the eFlash GUI, upon which eFlash creates a PPT presentation consisting of slides that alternate between the given sound files (or sections of them) and their corresponding answers. This PPT file is put into a user-designated directory for the user to retrieve.

Interview 2:
Interviewee name: Summer Chrisman
Notes:

Summer's prioritized feature list (the 20 second summary recap)

- Input your own words and definitions
- Full screen flash cards, sequence of word-def-word-def
- Full screen flash cards, fill in the blanks/multiple choice/provide own answer
- Intelligent search of your own database (find-as-you-type)
- Annotate your own pictures, make testable as well
- Annotate your own audio files, make testable as well

Full interview transcript with Summer Chrisman

- Instead of a new window every time I want a new window, here's a word, a definition, a sentence, and any weird rules it has (notes)
- Would be used for Swahili studying
- Just open up a program and "do it"
- Like when you upload a new piece of music, fill it in all at once and it's all there (taking an example from mp3 management)
- Select which category to use as a quiz (definition, the word, the sentence (?))
    - Full screen = less distraction
    - Food, vocab (every set = a quizzable list) (like, a week, transportation, etc)
- Problems with cards
    - Unwieldy
    - Can see through them to see the answer sometimes
    - Pain to organize on your own
- Good things about cards
    - Forcing yourself to write it down
    - See and correlate the visual definition of the word
    - Say it as you do it
- If I don't remember a word, press a button and get a sentence
- Move with keys (clicking is slower than pushing), keep up for however long it takes for you to get it
- Want to see the definition before you move on (a default option?)
- Using multiple choice
    - Might be hard ☺ she recognizes this at least
    - List of words and list of definitions – mix and match pairs
- If I have the option to tag audio to it (record my own voice), then auto-listen to it
- No need for pictures – one font, black letters, white background: simple = good!
- What's hard about learning languages? The memorization required

- Searching word database (type as you find – i.e. Firefox, cell phone auto-complete or in browsers, when you type the URL then it starts to pop up)
- Could we use this for biology? Any terms, any explanations you want at all
- Embedded links (I could input my own pictures but might be tricky) – put in molecules, chemfinder (just have it in the help or something as a reminder)
- If I have a picture from biology, click a point in it, write something associated with a point, and when I mouse over it, you'll see a pop-up for the definition of the point (and cross-link – think like stickies in Google Earth)
    - ex. Use for the anatomy (lungs, heart, cross-reference definitions)
- Test again in flash cards, or flash pictures with active participation ("pop up a blank box or something where I have to identity what it is")
- If you were doing history, we could add annotatable timelines
- "New…" – what? Make templates so it's easier to figure out what you want
    - "Here's a history file, here's a whatever – let me add to whatever I already have without having to start from scratch again"
    - So remember your most common actions, "open up course #101…"
- Full screen but leave the stupid little window bar at the bottom
    - (So I can get out of it still) – i.e. maximized screen

Use cases:

- The user opens eFlash and navigates to the "Create new flashcard" screen. She then either selects a preexisting template or creates a blank flashcard, causing the screen to reflect her choice. This screen allows her to input all of the information of the flashcard in a single window. After she inputs her information, with the click of a button, the flashcard is created and stored in the database. The user is then presented with the option of creating another flashcard or returning to the main menu.

Interview 3:
Interviewee name: Rocio Ramirez
Notes:

- Used them when first started learning Portuguese
- Color is important – learn the word that you know on one side, and then on the other side in a different color the definition
- A list of small words that I don't know – two columns on one flash card, color each side differently
- Multi-colored flash cards don't work for me, white works fine, I add my own color because the rest would be much more difficult
- Split them into category based on semantic meaning (kitchen, living room)
- Quizzing would be helpful (multiple choice?)
- Sound of word also important – recording my voice would also be nice, so I can hear my own intonations (learn how to hear? accents difficult to pick up on)
- Rules-based for accents, I could practice my own
- Write one word and then all the tenses come after it (save it all on one card, instead of all the different tenses on different cards)
    - Example quiz – with the word, you have to fill in the different tenses
- Do you need mixed media? Pictures, not for me

- Economics? Not really... Math? Not really… just language based (or memorize)
- A problem of thinking in multiple languages – instead of doing two translations for a word, I would need possibly more than two (three or four "sides" to a flash card) – especially helpful to see connections between the different languages

Use cases:

- The user opens eFlash and goes to the flashcard-creation screen. Here, she is presented with a variety of tools to customize the layout of the flashcard, such as choosing text color, moving text locations, etc. She also has the option to add additional sides to the current flashcard. Once she is finished with inputting the data, she clicks a "Save" button, storing the flashcard into the local database. Finally, she is presented with the choice of creating another flashcard or returning to the main menu.