Daryl Pham 5185566
Brandon Starler 4877361
Eric Melton 4724746
Wei Jia Jemes Li 4872040

# FlashCards

**Overview:**

The application we have decided to create is a flash-cards application that allows for users to create and use flash-cards to study for any subject they desire. We chose this because it is a very simple idea, does not take too much explanation, and has minimal functionality. Because this application allowed for functionality that was not too over-bearing, it ultimately allowed us to become very flexible with interface design that was easy for the user and did not take too much away from the application's functionality.

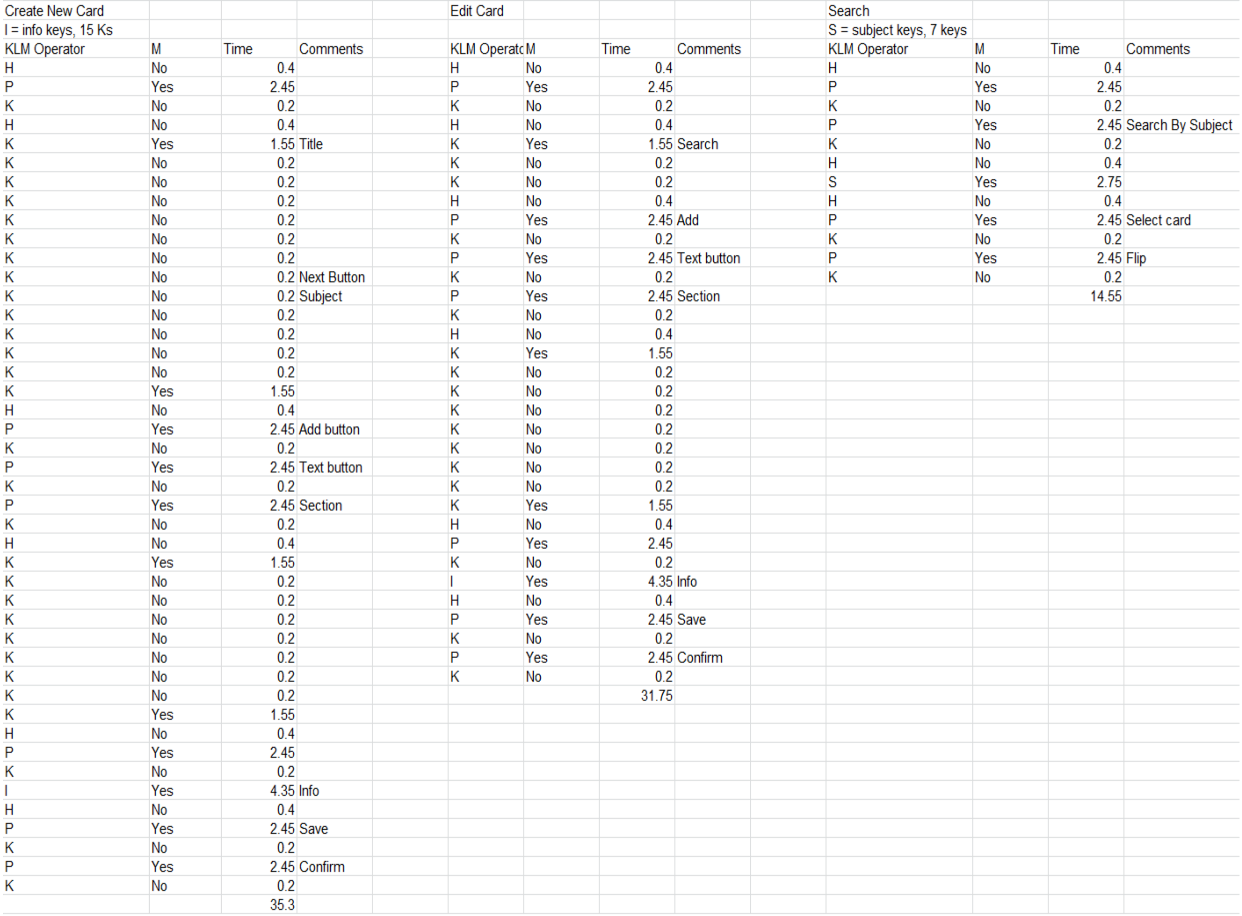
This application would support three main functions, the ability for the user to make new flash cards, edit flash cards, and search through his/her database of cards. We will be presenting, within the documentation, the many interface changes we went through as iterations passed. Each main function was tweaked in order to allow the best possible experience for the user.

**User Analysis:**

Target population for this application includes male and female students between the ages of 12 and 25. This application will be used for the purpose of studying so those who download the application will most-likely be attending school. It will also have the capability of being used for other purposes such as recipes, cheat codes, etc. Application will also handle universal usability allowing for novice users to be able to share the same experience as expert frequent users.

**Task Analysis:**

The task of this application is to allow anybody with an android smart-phone to have access to flash cards if they may need it. We aim to make the process of making, editing, and using flash cards with application to be easier than physically making and using flash cards; if it can do this, we have succeeded.



| Create New Card | | | |
|---|---|---|---|
| I = info keys, 15 Ks | | | |
| KLM Operator | M | Time | Comments |
| H | No | 0.4 | |
| P | Yes | 2.45 | |
| K | No | 0.2 | |
| H | No | 0.4 | |
| K | Yes | 1.55 | Title |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | Next Button |
| K | No | 0.2 | Subject |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | Yes | 1.55 | |
| H | No | 0.4 | |
| P | Yes | 2.45 | Add button |
| K | No | 0.2 | |
| P | Yes | 2.45 | Text button |
| K | No | 0.2 | |
| P | Yes | 2.45 | Section |
| K | No | 0.2 | |
| H | No | 0.4 | |
| K | Yes | 1.55 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | Yes | 1.55 | |
| H | No | 0.4 | |
| P | Yes | 2.45 | |
| K | No | 0.2 | |
| I | Yes | 4.35 | Info |
| H | No | 0.4 | |
| P | Yes | 2.45 | Save |
| K | No | 0.2 | |
| P | Yes | 2.45 | Confirm |
| K | No | 0.2 | |
| | | 35.3 | |

| Edit Card | | | |
|---|---|---|---|
| KLM Operatc | M | Time | Comments |
| H | No | 0.4 | |
| P | Yes | 2.45 | |
| K | No | 0.2 | |
| H | No | 0.4 | |
| K | Yes | 1.55 | Search |
| K | No | 0.2 | |
| K | No | 0.2 | |
| H | No | 0.4 | |
| P | Yes | 2.45 | Add |
| K | No | 0.2 | |
| P | Yes | 2.45 | Text button |
| K | No | 0.2 | |
| P | Yes | 2.45 | Section |
| K | No | 0.2 | |
| H | No | 0.4 | |
| K | Yes | 1.55 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | No | 0.2 | |
| K | Yes | 1.55 | |
| H | No | 0.4 | |
| P | Yes | 2.45 | |
| K | No | 0.2 | |
| I | Yes | 4.35 | Info |
| H | No | 0.4 | |
| P | Yes | 2.45 | Save |
| K | No | 0.2 | |
| P | Yes | 2.45 | Confirm |
| K | No | 0.2 | |
| | | 31.75 | |

| Search | | | |
|---|---|---|---|
| S = subject keys, 7 keys | | | |
| KLM Operator | M | Time | Comments |
| H | No | 0.4 | |
| P | Yes | 2.45 | |
| K | No | 0.2 | |
| P | Yes | 2.45 | Search By Subject |
| K | No | 0.2 | |
| H | No | 0.4 | |
| S | Yes | 2.75 | |
| H | No | 0.4 | |
| P | Yes | 2.45 | Select card |
| K | No | 0.2 | |
| P | Yes | 2.45 | Flip |
| K | No | 0.2 | |
| | | 14.55 | |

To better understand the tasks that the user will do, we did a KLM to see how long it takes a user to use key features of the program. We realize that a KLM is for expert users only. We hoped that it would still give us an accurate time even if users weren't experts at using our program because most users would be knowledgeable with flash cards in general. Our KLM showed that in order create a new card, it would take roughly around 35.3 seconds to input one card, 31.75 seconds to edit a card, and 14.55 seconds to search for a card. Before we optimized our program, it took around 12 extra seconds when searching if you had multiple subjects. This is at the point where we started to add auto-completes which sped up the time.

**Documentation:**

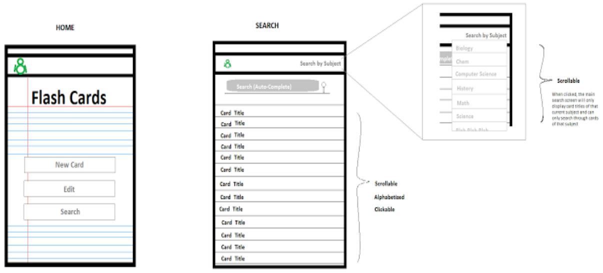
1.

For the first iteration, we decided to make a very simple home-page layout that had only the name with three well-spaced buttons under it. These buttons would be "New Card", "Edit", and "Search".

Upon pressing the "New Card", the user is taken to a new page with a mandatory EditText field to add the title of the card, and a second EditText field to add the subject of the card. At the bottom of the page is an "Add" button where the User can add more text-boxes with information he/she wants to put in. To save changes of the card or cancel changes to the card, there is are three options "Save", "Cancel", and "Clear All". Upon hitting "Save", the user is taken back to the main menu page and the card is saved with its chosen card title and subject. Upon hitting “Cancel”, the user is taken back to the main menu page and the card is not saved. Finally, clicking “Clear All” keeps the User on the same page, but every text field that was entered in is deleted.

The second button, “Edit”, the user is taken to a page that has a search bar at the top of the page and a list of alphabetically organized cards under it. Within this page, the User has the option of entering a card title in the search-field or typing in a Card title to narrow the list. It was important for us to utilize universal usability - allowing novice users to use the scrollable list if they did not feel comfortable using the search bar. We also gave the search bar the auto-complete feature giving expert users the ability to get the card they wish to access, faster. When the Card title is clicked, the application will bring the user to a new page similar to the page in the “New Card” button, except the text fields will be editable. When the user is finished, the user can select “Save”; “Cancel” which will bring the User back to the main menu page, making no changes to the card; and “Clear All”, keeping the user on the same page will all text fields cleared.

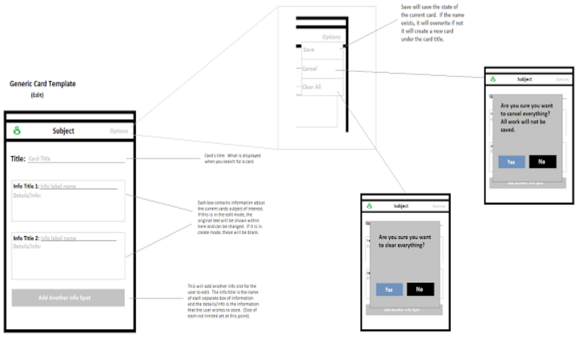
The third button, “Search” would take the User to the same page as the “Edit” had and would list the cards in alphabetical order by Card title and have a search bar at the top of the screen. Within this page, we wanted to make sure there was an accessible list for the user to be able to scroll through if they were novice users, and also an auto-complete search bar if the user was an expert. When the User selects a card, the card name, along with its subject and information will be shown.



When clicked, the main search screen will only display card titles of that current subject and can only search through cards of that subject

To-Do

Learn how to save these items
Learn how to add more fragments to the activity with a button click (for the "add" option in the edit/create mode)
Learn how to list all the items and display the list and make each item clickable with the path to the specified item
Select subject when creating/editting card
Allow for user to create a new subject


Generic Card Template
(Edit)
Subject
Options
Title: Card Title
Info Title 1: Info label name
Details/Info:
Info Title 2: Info label name
Details/Info:
Add Another Info Spot
Options
Save
Cancel
Clear All
Subject
Are you sure you want to cancel everything? All work will not be saved.
Yes
No
Subject
Are you sure you want to clear everything?
Yes
No

Save will save the state of the current card. If the name exists, it will overwrite if not it will create a new card under the card title.

Card's title. What is displayed when you search for a card.

Each box contains information about the current cards subject of interest. If this is in the edit mode, the original text will be shown within here and can be changed. If it is in create mode, these will be blank.

This will add another info slot for the user to edit. The info title is the name of each separate box of information and the details/info is the information that the user wishes to store. (Size of each not limited yet at this point).

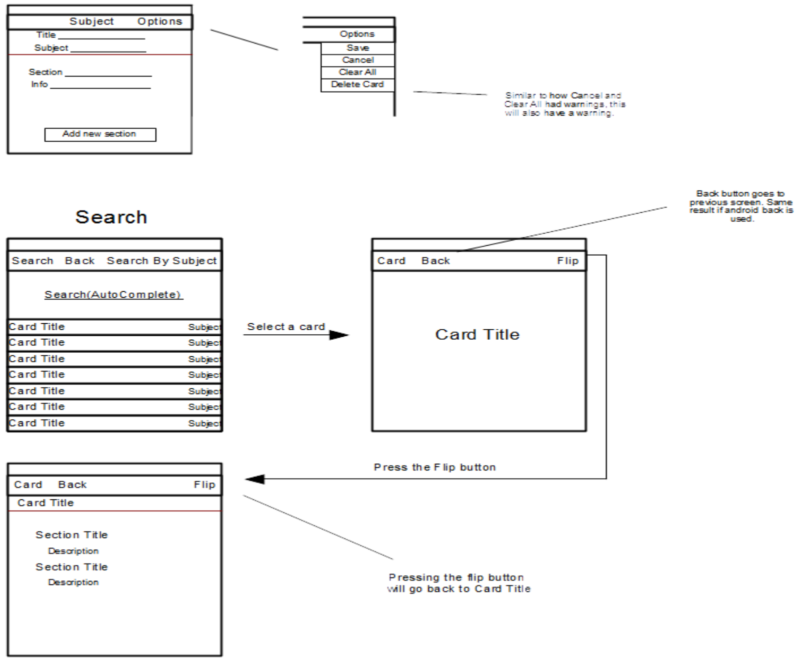
2.

For the second iteration, we tested the prototype of our application and found slight inconveniences that could hinder use usability and made a few fixes.

The first new feature we implemented within this iteration was the “Back” button. Although there is a “Back” button on Android devices already, the team felt that it would easier for novice users to have a “Back” button on the application itself. When clicking the buttons “Edit” or “Search”, there was no option for the user to return to the page they were previously on. The “Back” button would now be placed in the action bar in the top left-hand side; it would be on the Search page and the Card page so the user could return to the main-page and the search-page, respectively. The “New Card” and “Edit Card” pages have a Save and Cancel button so we felt the “Back” button was not necessary.

The second new feature we implemented within this iteration was the addition of a new option within the “Edit” layout. When the user presses the “Edit” button, finds the card he/she wants to edit and are unhappy with the card, they are able to click the newly added “Delete Card” option. The “Delete Card” will now allow the user to completely erase a flashcard from the database.

Another feature we added into the application was the “Flip” function. When the user accesses his/her card from the “Search” layout, and the card is displayed, we originally had the title with all of its information on one side of the card. Real flash-cards generally have the title and subject on one side with the information on the other side. We wanted a more immersive experience for the User, allowing them to feel as if they were using a digital flashcard, so we implemented the “Flip” function that would make it so the card title and subject would be on one side of the card while the information would be on the other side, accessed using the Flip button.



Similar to how Cancel and Clear All had warnings, this will also have a warning.

Back button goes to previous screen. Same result if android back is used.

Pressing the flip button will go back to Card Title

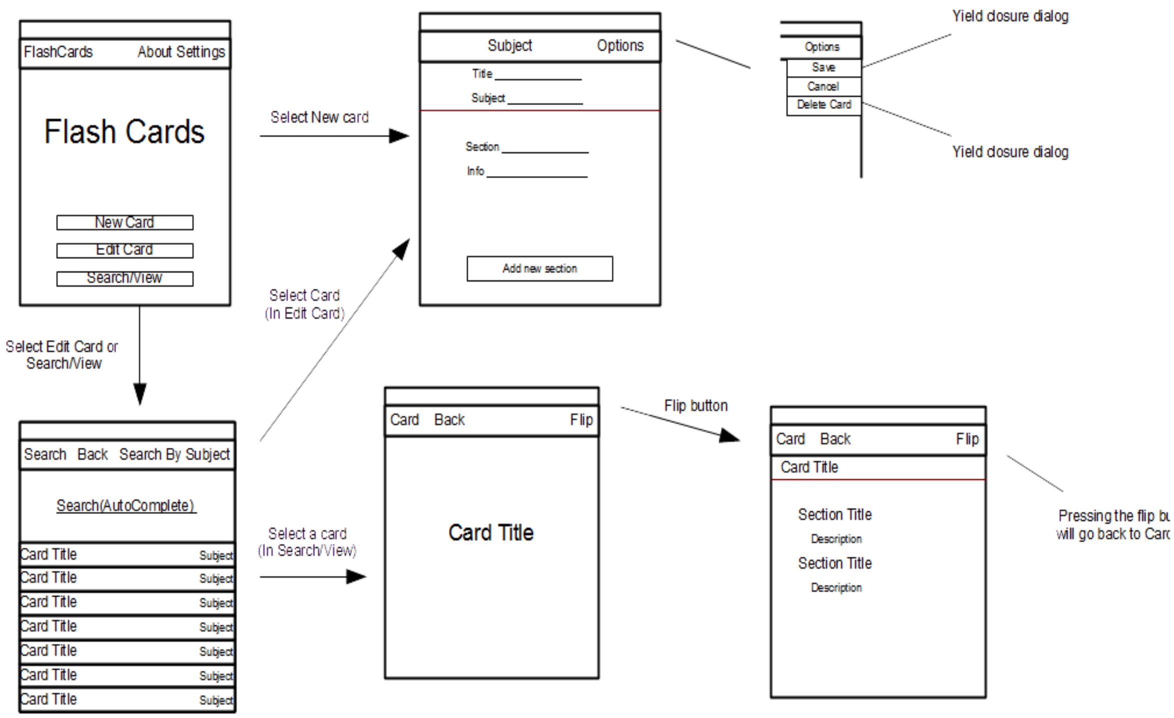
3.

For the third iteration, we again, tested the prototype of our application and looked towards making better use of the principles and guidelines of UI Design to allow for the user to have a slightly more comfortable experience. To do this, we found two areas that we could improve, and another additional feature that could be added to help for usability.

The first feature was the removal of the "Clear All" option. We decided to remove the "Clear All" option because we felt this button was too similar to the "Delete Card" option. If the user chooses to clear all the data that has been entered, it is nearly the same thing as deleting the entire card itself. Looking back on Hick's Law, the more alternatives the user has to make, the longer it takes for the user to make a decision. With the deletion of the "Clear All" option, the user has one less alternative choice to make.

The second feature was the addition of a dialog to yield closure. When the user is entering information into a card or editing information, they will eventually have to save the data, or choose to cancel or delete the card. When the user is finished with the card and selects one of the options, they are prone to error; with a single mis-click, the user could possibly click "Delete Card" when they actually meant to click "Save". To prevent this, we added an additional dialog once one of these options are pressed to ask whether the user is sure with their option.

The last feature we added was part of the "Search by subject" function. When the user clicks "Search by subject" they enter a separate dialog that lists the currently created subjects, unfortunately there was no way for the User to go back to the page with all cards listed. Because of this, we added an "All cards" button at the bottom of the "Search by Subject" page. If the user is in the "Search by Subject" and wants to return back to the page search page with all cards listed, he can do so by clicking the "All cards" button.



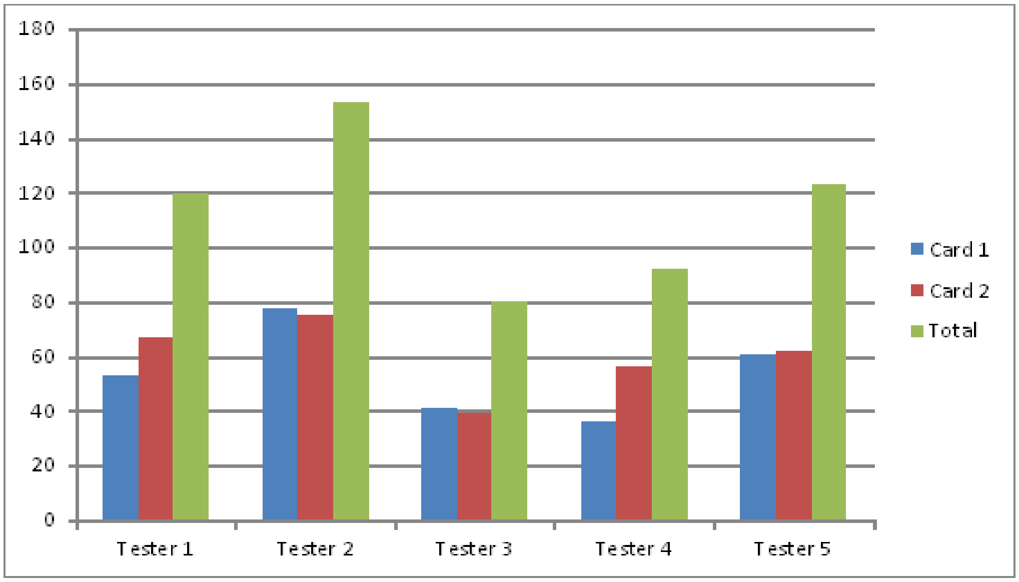
**Evaluation Results:**

The questionnaire was given to five friends in our household. Each user was given this questionnaire at random times when their schedules permitted due to finals. To do this test, we gave each of them the same mobile device, the phone we used for development, and documented the time it took them to follow a certain set of instructions. We also asked them in the post-questionnaire what suggestions they might have had to make the application better; and we also asked them if they preferred to use our application over physical flash cards.

To do this test, we handed each of the five members two flash-cards each, being the same for all five members. They contained a card title, subject, and two section titles and descriptions. They are told to create these flash cards, edit the name of the card title, then told to view each of the two cards front & back one at a time. They were given no suggestions to use the application, but they were given the opportunity to familiarize themselves with the application so we could have fair results.

First Card:
Title: George Washington edit to: Jorge Washington
Subject: History
Section Title: President
Info: 1789-1797
Section Title: Background
Info: Born in Virginia, and was a founding father
Second Card:
Title: Pythagorean's Theorem edit to: Pythagorean's Law
Subject: Math
Section Title: Equation
Info: a^2+b^2=c^2
Section Title: Variables
Info: a = length, b = height, c = hypotenuse

The results from the questionnaires at the bottom of the page are on the separately attached Excel sheet.



Y-axis represents time in seconds

**Problem Scenarios:**

A.

Sarah has been attending UCSB for her first quarter and quickly realizes that she will need a better means of studying in order to do better in her classes. Luckily, she recently downloaded the cool, new Flash-Card app that allows her to create and use flash-cards for studying purposes. She finds out that she is having a midterm in her art-history class next week, which gives her just enough time to create flash cards early so she can use these to study before the test. She opens the application, clicks “New Card” to begin, there, she enters the title of the card she desires, chooses the subject she intends to store it as, and clicks the “Add” button to add a new text box to write her information. She feels the need to add another text box so she clicks “Add” once more. The flash card now has the title, subject, and two text fields filled with information she intends to study. She clicks the “Save” button in the actionbar that takes her to a dialog to confirm her decision, and clicks Yes. This saves the flash-card, which sends her back to the main page of the application. She decides she wants one more flash card. She clicks “New Card” once again, makes a card with a new title, selects the same subject, and clicks “Add” to add one new text box. She then decides she actually does not need this card. She goes up to the actionbar and selects Cancel. A dialog comes up asking if the user is sure of their decision, and she clicks Yes. She is back to the main-page.

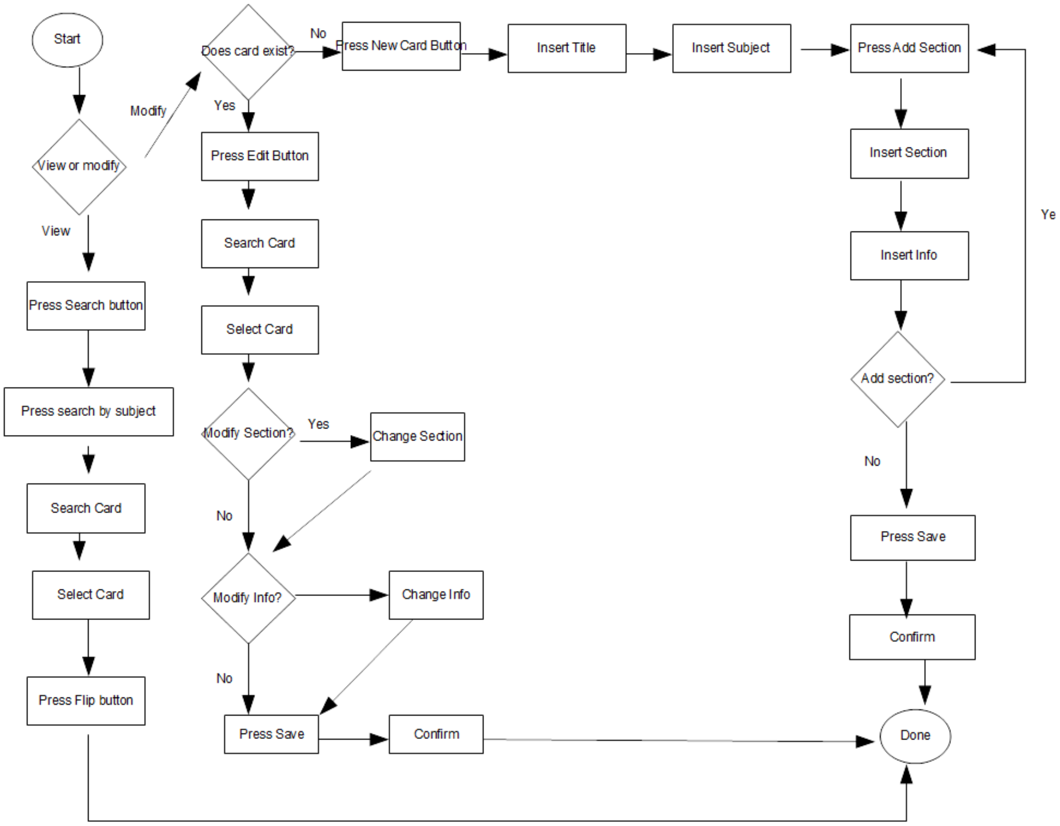
B.

Sarah has been attending UCSB for her first quarter and quickly realizes that she will need a better means of studying in order to do better in her classes. Luckily, she recently downloaded the cool, new Flash-Card app that allows her to create and use flash-cards for studying purposes. She has a midterm in her art-history class within the next couple of days so she is planning to look through the flash cards she has already made since getting the application. She opens the Flash-Card application and clicks “Search” to begin her studies. She sees a list of flash cards organized in alphabetical order by card title and wants only to see the list of cards that contain information regarding art-history. She clicks the “Search by Subject” button in the actionbar and clicks “Art-history”. There, she is able to see the list of flash-cards only dealing with Art-history. She clicks the first flash card, looks at the title, and realizes she does not remember facts pertaining to the card; she then clicks “Flip” which brings her to the other side of the card - listing the information. Once she is finished studying the card, she clicks “Back” returning her to the list of flashcards within the art-history subject.

C.

Sarah has been attending UCSB for her first quarter and quickly realizes that she will need a better means of studying in order to do better in her classes. Luckily, she recently downloaded the cool, new Flash-Card app that allows her to create and use flash-cards for studying purposes. She has a midterm in her art-history class within the next couple of days so she is planning to look through the flash cards. Her professor has discovered an error in their notes about a particular painting, this causes Sarah to need to go back and make a change with a flashcard she has already made. Fortunately for her, the FlashCards application she has on her phone lets her go back and edit information inside of cards she has already created. Using the “Search” function, she finds the card in question, she then finds the correct text view that contains the wrong information and clicks it. This prompts the application that she wants to edit the text that is there and brings up an editable field for her to change, remove, or add whatever she likes without having to make an entirely new card from scratch. When she is finished, she clicks “Save”, returning her back to the previous search page.

**Use Case:**



## Summary:

In summary, we were worried we put too much thought into our first iteration and that we would run out of functionality and interface ideas to implement, but this was not the case.We applied many of the eight golden rules to our application and utilized the two basic laws in User-free usability inspection methods to make for a user-friendly interface.

To begin, we focused on the  Laws of interface design such as use of both Fitts' Law and Hick's Law. For our buttons, we paid careful attention to make the buttons aesthetically pleasing by being small enough, but large enough as well, for the user to have an easier time pressing buttons, utilizing Fitts' Law. For utilizing Hick's Law, we came to a point in our application where there were two options "Delete Card" and "Clear All" that had almost the same functionality. We decided to remove the "Clear All" option so the user had one less alternative to choose from, making it easier for the user to make a decision.

As for the eight golden rules, we tried to focus on each rule as we progressed through iterations. We made an effort to have a consistent interface throughout the main-page and other pages so the user did not get confused. We also catered to universal usability by including list-menus and ease-of-use for our novice users,while also adding an auto-complete feature for our expert users. We also added a feature into our application that would ask the user if he/she was sure after clicking "Save" or "Cancel", these were meant to offer feedback to the user and give the user a sense of closure that the particular action he/she was doing is now over. This helped to prevent errors, as users now had a second-chance before committing to an action. For the sixth golden rule, in the second iteration of our design, we added the “Back” button to permit easy reversal of actions; although the android mobile device already supports one on the hardware itself, we felt it would be easier for the user to have one on the application. To support internal locust of control, we focused on allowing the user to have the easiest time to create, edit, and search through his/her cards. We implemented options that would allow the user to feel as if he owns a virtual set of flashcards. Finally, for reducing short-term memory, when we made the search function, we listed the cards in alphabetical order so the user could scroll through the cards and find his/her card, we also allowed for search by subject which gives the user less to have to remember, and an auto-complete feature for the search bars to help the user remember his/her card title.

Pre-Study Questionnaire

Thank you for participating in this simple user study. Your task is to create two flash-cards in two separate subjects, save them, and open them up using the built-in search function. We will time how long it takes you, so please enter the information that we give you as fast as you can while maintaining accuracy. You may familiarize yourself with the application before you use it so we get fair results.

1) My age:

2) My gender (m/f):

3) I own a smart-phone (yes/no):

4) I am familiar with how flash cards are used (yes/no):

5) I have experience with smart-phones, including usage of entering data, navigating interfaces and handling requests.
(Please answer with a number from 1…7, where
1 == I am unfamiliar with these concept
…
7 == I am an expert at these concepts)

Post-Study Questionnaire

Thank you again for participating in this user study. To conclude our data collection, we would like your opinion on the interface of our application.

1) How did you feel about our interface, was it usable?
(1: Thought the interface was hard to use and affected my ability to enter data
…
4: Thought the interface was not too great, but able to be worked through
…
7: Thought the interface had everything I needed to allow for smooth data entry)

2) How easy was it to make a mistake while trying to follow the instructions specifically because of the interface?
(1: Thought it was hard to make mistakes with this interface
...
7: Thought it was easy to make mistakes with this interface)

3) Would you rather use our application to make flashcards, or would you rather use hand-made physical flashcards?
(1: Would only use hand-made physical flashcards
...
7: Flashcards application is the only way to go)

4) Did you notice anything in particular that could have been done differently on the interface that would have made the experience smoother?

References

http://stackoverflow.com/questions/4580664/how-put-spanned-into-list-setadapter

The link above is where I got the basic idea on how I would go about overriding the ArrayAdapters I used in the project. I created a separate class that was used to store the name and subject of each card that I would list in the ListViews or AutoCompleteTextView. However, in order to list these classes as part of the Views, I had to override the ArrayAdapter to store and use my class as input.

http://stackoverflow.com/questions/2208225/android-how-do-i-overwrite-the-filter-for-my-arrayadapter

The link above is where I learned how to sort my new ArrayAdapter to list items that I search for in the AutoCompleteTextViews. To do this, I had to override my filterable class inside the ArrayAdapter. Once I do that, I can change the way things are sorted.

http://developer.android.com/reference/android/widget/ListView.html

I learned how to use ListViews from the android developer site linked above. I used the ListView in the search page where the background lists the different items. I added the above adapter technique to be able to get that functionality.

http://stackoverflow.com/questions/6674341/how-to-use-scrollview-in-android

In the link above, I used the XML suggestions there to create a ScrollableView that I used in the Edit and Create Card Views. Each of these Views add text input Views dynamically when the user requests and because of that, I needed something that could scroll. This is where I found out how to do that.