**Title**: Study in a Flash

**Problem**: Students need to study for exams. A good way to do this is to put notes into flashcards, for quick review. Instead of handwriting flashcards, we can create flashcards on the computer!

**Primary stakeholder**: The primary users of this program will be students studying for exams who need to create flashcards and then review the flashcards to study off of

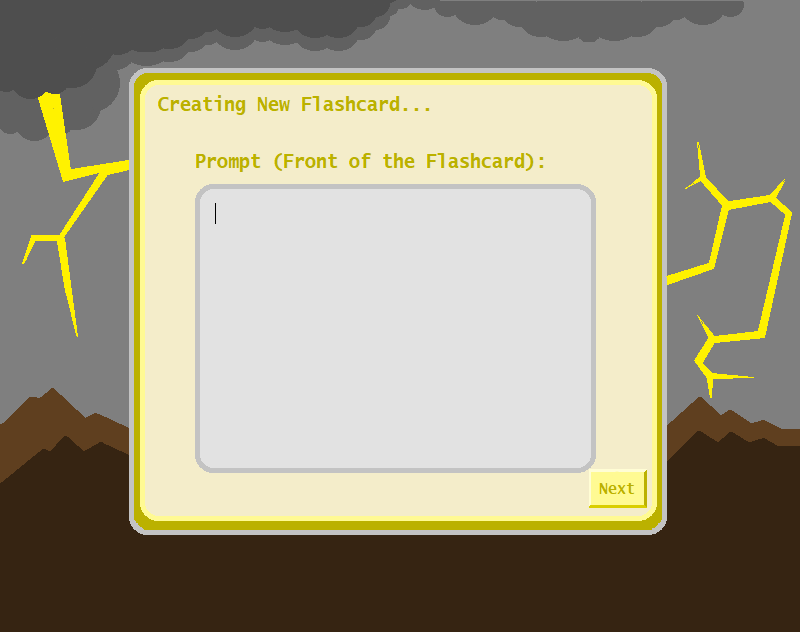
**Graphical User Interface**:

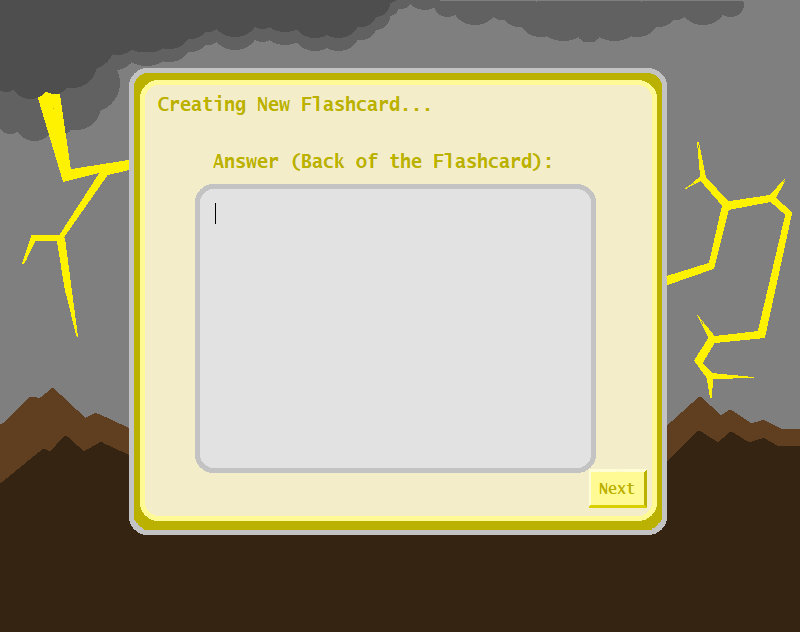
At the load screen, users can:

1. Load from pre-existing set of flashcards (tab delimited .txt file, with each ‘flashcard’ preceded by a unique numerical key)
2. Create a new set of flashcards



If users select “New”, they will proceed directly into flashcard “Create Mode”:





If users select “Load”, they will have an option of entering into “Create Mode” as above, or “Study Mode”, which will allow them to flip through a randomized ordered list of the flashcards:





Study mode will allow you to flip through the loaded or created flashcards:

* It will randomize the list
* It will show the front first
* Upon clicking, it will flip the flashcard around

Create mode will prompt for the front of the card, and then the back of the card:

* It will have an option to save all, and move to study mode
* Exiting out of create mode will save to the .txt file
* Users will have the ability to save the file with a specific name, like “CS400 Midterm Study”

Upon entering study mode or create mode with a pre-loaded file, we will load the flash card file and store it in a data structure as below.

**Data**:

We will be loading the data from a pre-existing set of flashcards (tab delimited .txt file, with each ‘flashcard’ preceded by a unique numerical key). Or, if jumping directly into study mode from create mode, we will be storing the new flashcards in the below data structure as well as saving off into a .txt file.

* Each flashcard’s key is unique, so we should be able to achieve O(1) retrieval time if we use a hash table
* We’ll do the following:
  + For every flashcard stored in the .txt file (if we are pre-loading a list of flashcards), we will create an object and store it in a hashtable.
  + We will separately maintain a sorted list of the unique keys in a LinkedList.
    - We should only need to sort it once upon loading the file
    - This list will have a method nextID(), which will return the last ID + 1 to automatically generate a unique key for the newest flashcard
    - This method will be used when creating new flashcards, so we can avoid having to resort this list as new cards are created
  + Every flashcard will be represented as a Flashcard object with the following basic properties:
    - Key = this will store the unique key of the flashcard
    - Front = this will store the contents of the front of the flashcard as a string (this can be considered the flashcard prompt)
    - Back = this will store the contents of the back of the flashcard as a string (this can be considered the flashcard answer)
  + We will run the key through a hash function, and create a hashcode for each Flashcard object
  + Using this hashcode, we will create a hash index and then store the Flashcards away in the hashtable.
  + If we create enough flashcards, we will need to rehash the table

Once the file is loaded as above in Study Mode, we will then:

1. Copy the sorted LinkedList of keys into a new LinkedList in randomized order
2. Iterate through this randomized List, using the hash function on the key to retrieve the Flashcard object in O(1) time
3. Present the front of the retrieved Flashcard object to the user
4. Upon clicking the mouse, present the back of the card to the user
5. Upon clicking the mouse again, move on to the next Flashcard