Project Proposal for Memorier

Project Code:

- H20

Our Team:

- Zhang Jiekai (jzhanger@connect.ust.hk)
- Zhang Zhong (zzhongfc@connect.ust.hk)

Project Background and Description

“Preface”: In such a fast-paced world, especially for our students, we have lots of things to learn and remember.

Have you ever suffered in memorizing things when you are learning a new language, or you are preparing for an examination? Have you ever experienced studying all day but it turns out you only remembered the most recent things before the final examination?

To prevent this from happening, we are coming with **Memorier** , a useful tool that helps you to remember to summarize and to test yourself!

Memorier is a tool using **memory cards** to help you memorize things like words, questions, important points at **anytime, anywhere**. And it also allows you to set up a **daily/weekly test** for yourself to enhance your memory and find out the cards that are unfamiliar or unskilled, and it will often **push unfamiliar knowledge to you** until you fully grasp the points!

Expected End Result of the Project

Software

The result of the project i.e. the software should be able to fulfill user’s demand for managing learning materials. It should be like a simplified version of OneNote (only plain text) plus some student-based function.

1. Testing
    - Auto-generated daily and weekly tests.
    - it can automatically change its content base on the user’s familiarity with cards.
2. Reciting
    - Based on the user’s input format, extract important sentences.

    - Choose numbers (remembering history years etc.)
    - Based on algorithms, automatically choose the important words. (more discussion needed)

3. Words auto-translation

    - Calling online APIs, display words translation, and formation.

4. Cloud synchronize

    - Find your card everywhere, anytime.

Programing

Making an efficient portable database manager (more discussion needed). It should be able to dynamically adding, deleting, and searching cards. With each operation using less than O(log) time.

OOP / Data Structures / Other Techniques

OOP Techniques

Currently, we plan to use three classes (there should be more for the final program):

- **“Card” class** for processing memory card (Different memory types are inherited from this)
- **“Test” class** for processing tests and its result
- **“CloudSync” class** for dealing with information cloud synchronization, data parsing.

Data Structures Techniques

- For general storage: Double linked list. Writing and reading files block by block.
- For reciting test: TF-IDF algorithm / TextRank
- For database: AVL tree (splay tree)

Other Techniques

- **MySQL database** for storing our cards
- **Server-side programming**, we wish to have a server for storing the data
- **HTTP related techniques** for cloud synchronization and some APIs to lookup dictionary
- **Qt** for the user interface (we primarily develop Windows, while we hope it can be ported to Android in the future)

External Libraries to be used

- Qt for interface (and porting to Android)
- HTTP protocol related
- MySQL related
- Some encryption algorithm for storing date like password

References

1. Anki: A memory card software, which we found is very similar to our project, but it is sort of hard to use for beginners. Our goal is to realize some of its basic functions, add more useful functions for college students, and optimize user experience as much as possible and develop a more satisfying interactive pages.
2. Qt: For UI developing.
3. C++ MySQL Lib: We use this to implement the cloud sync.
4. C++ Http Protocol: For the server-end program. It will grab data for the database and sent it to the client.