

(WriteDown)

In-class translation app

Project Team: OJBK

Ruifeng Zhu

Mengmeng Liu

Yangling Hu

Yiheng Jiang

Ruifeng zhu: work for the Android interface
Yangling Hu: work for the database
mengmeng Liu: work for the AI handwriting recognition
Yiheng Jiang: work for the api stuff.

Table of Contents

1. Project Definition (100 - 200 words) – *Group responsibility*

- This application generally works for international students to take notes in class. According to the report, millions of international students come to America for study, for most of those students are suffering from the understanding of the classes, because of the unknown words. At this point, “write down” is the application can help them get to know the course clearly. The goal for “write down” is to use the original way like the notebook to allow students to take notes on the electronic devices, and in the meantime, it will recognize the handwriting by users and translate to their languages which can help students learn their course better. And also, these notes which users were created can be saved as their own account, the users can easily view edit, delete and download their notes all the time. Also, The purpose of share board is to allow users allow to their notes online for a better learning. This application will be available for the Android operating system first.

2. Project Requirements – *Group responsibility*

Functional User character :

- 1. Recognize the users handwritings get the value to use for the next step.
- 2. Use the words from step 1 and translate to the language which users want.
- 3. Save the notes to the account (cloud).
- 4. SignIn/Signup to allow users for more functions.
- 5. History board allow users to delete, download the notes.
- 6. Share Board allow users to share their notes with each others (Learning area).

User character:

There are two types of users interacting with the system: mobile application users and administrators. These two types of users use the system differently

They have their own demands. Mobile app users can translate text and form translation notes through handwriting recognition, and users can choose to upload their translation notes to the sharing board by logging in the account.

The administrator can manage the user's account, ban the account for illegal operation or screen and explain the illegal notes

Usability:

-UI: the UI interface will have a login and register function. It will have a username, password bar, and a register icon.

-Database interface: the interface system for a user account will need to be included a method to save information such as the words translated, and get another method to get the saved information

-Hardware interfaces: the application must take input from any generic touch screen in order for the user to navigate the menus and output to any current generic monitor compatible with Android cellphone.

Performance:

Write down project will allow at least 500 user login at the same time.

Writedown project will save client information after each transaction closure to prevent information loss because of the crash or loss.

System:

Developer language: java, SQLite

it will be a mobile application

Write Down project will require an Android operating system on a mobile phone with the minimum required specifications to run said operating system. For the database, the system works on SQLite.

Security:

SHA256.

1. Each account should have its own independent account and password, and the same account cannot be registered repeatedly

2. For malicious users, the administrator has the ability to directly delete and disable the user login.

3. Project Specification

- Focus / Domain / Area

For our application, we want to focus on the students who study abroad. Basic for the Asian students (Especially for Chinese international student).

- Libraries / Frameworks / Development Environment
Glide/ MVC / android studio.
- Platform (Mobile, Desktop, Gaming, Etc).

Basically for any Android devices both smartphones and tablets. (Apple software would be added later on.)

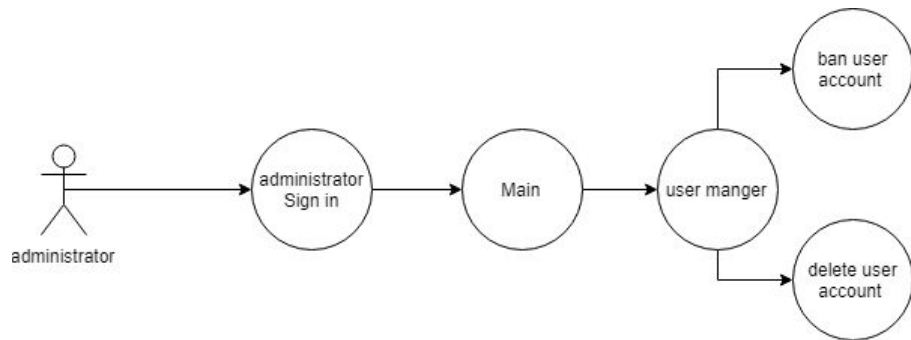
- Genre (Game, Application, etc).

It is a Multi-function note application. (reference, tool, efficiency).

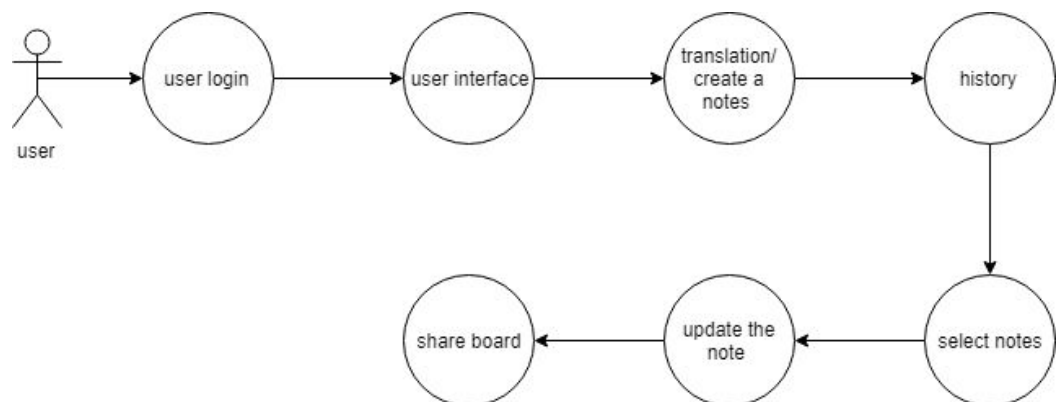
-----work on for the below-----

4. System – Design Perspective – *Group responsibility*

- Identify subsystems – design point of view
 - Illustrate with class, use-case, UML, sequence diagrams
 - Use case 1: the administrator manages the user account.
 - Actor: The user, The administrator
 - Prerequisites:
 - The administrator has an active administrator account.
 - Users have their own personal accounts
 - Use cases:
 - The administrator login the write down terminal and provides the valid username and password. The administrator then selects "user management" from the menu provided. The administrator then selects "Ban account" or "Delete account" from the user administration menu. The application then opens the administrator's account (user ID, user name) about the user. The application confirms that the account has been locked or deleted and asks the administrator to confirm. If the administrator selects "yes", the application will operate on the account. If the administrator selects no, the application returns to the default menu for user administration. From the menu, the manager then selects "back to main menu", which returns to the default menu option. The administrator can then select "exit" and the application will return to the default login screen.



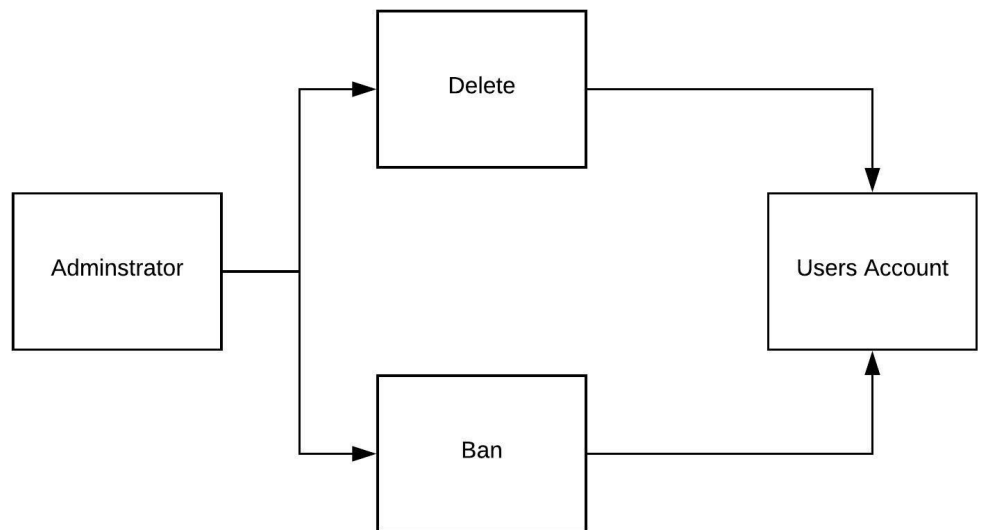
-
- Use case 2: the user uploads an existing history translation note.
- Actor: The user
- Prerequisites (one or more) :
- The user has an active account.
- There are translation notes in the user history
- Use cases:
- The user first logs in to write down and provides the correct and valid username and password. Then write down displays the default menu after logging in. The user then selects history from the default menu. Write down displays the user's most recent translation notes. The user then selects the required uploaded history translation from the history and clicks "upload". The system will ask the user to provide the title and edit requirements, and then click "ok". The system will upload the Shared document to the sharing board.



○

- Design choices (Optional)
- Sub-System Communication (Diagram and Description)
 - Controls

Administrator: the administrator can manage the users account. Like delete or ban the users account.

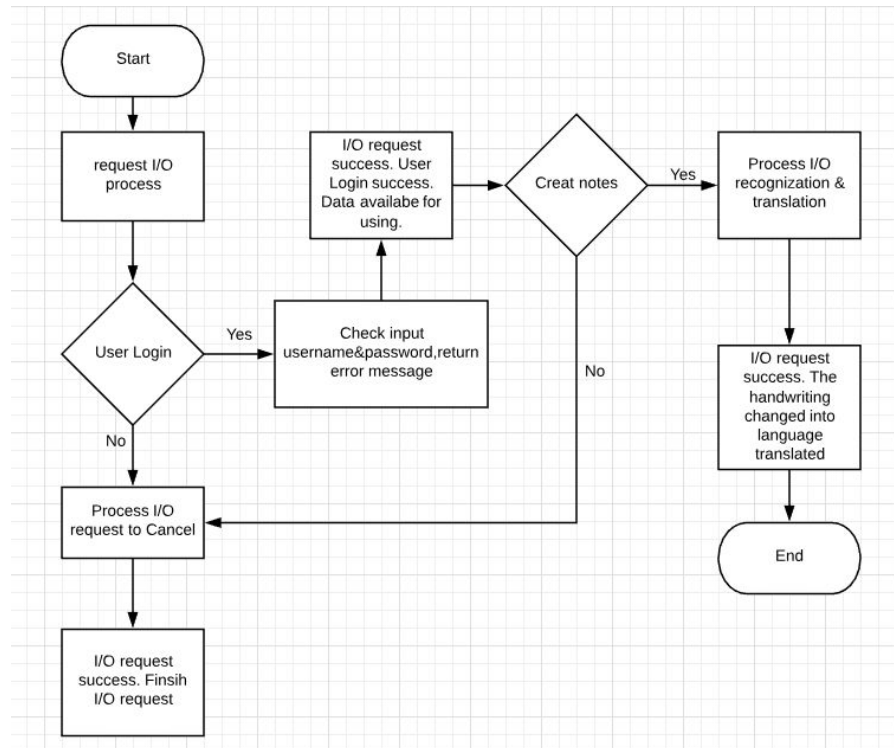


User: the user can use all the functions of the application. When user open the write down application, users can sign in, sign up, create notes delete notes, share notes.

- I/O (Yangling Hu)

I/O first to process the user login request, process “no” to process cancel request, process “yes” to check input username & password, return error message. Once the I/O request success, allow user to next step.

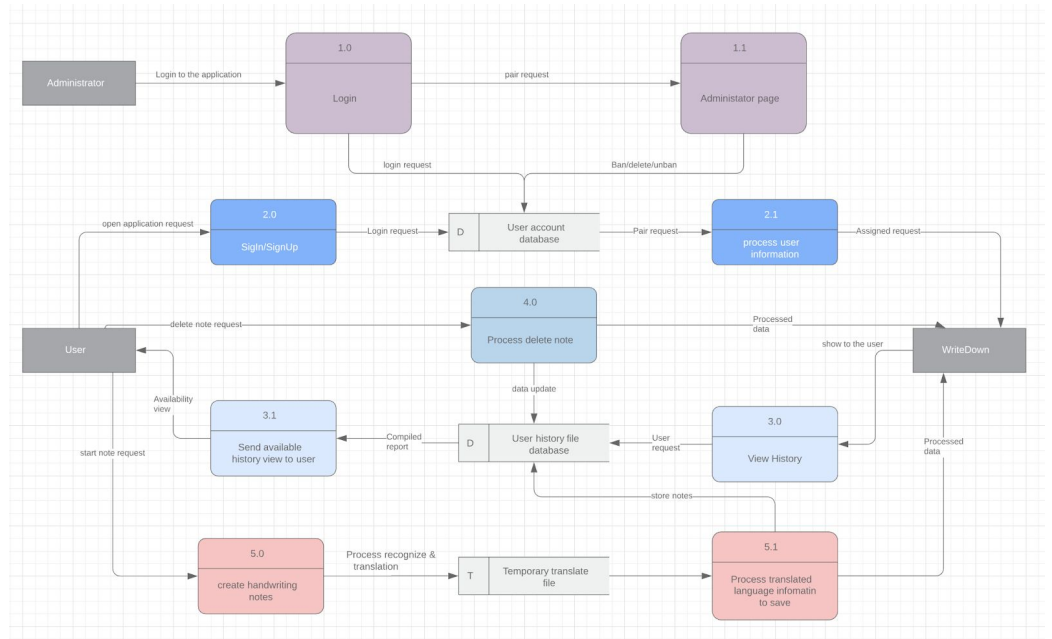
To create notes, process “no” to process cancel request, process “yes” to process I/O recognize & translation. Once the I/O request success, the handwriting changed into language translated



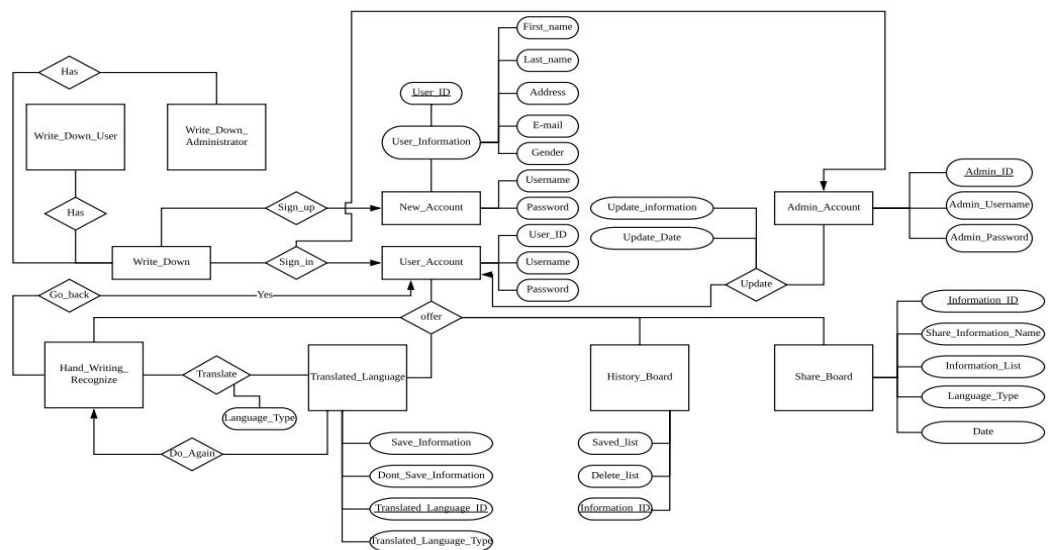
○ DataFlow (Yangling Hu)

1. User signIn or sign up the application it connect the login information from the user database to show and update users' profile. To create the new notebook, it saves from the history, and history connect to the database for updating the notes and delete in the user database.

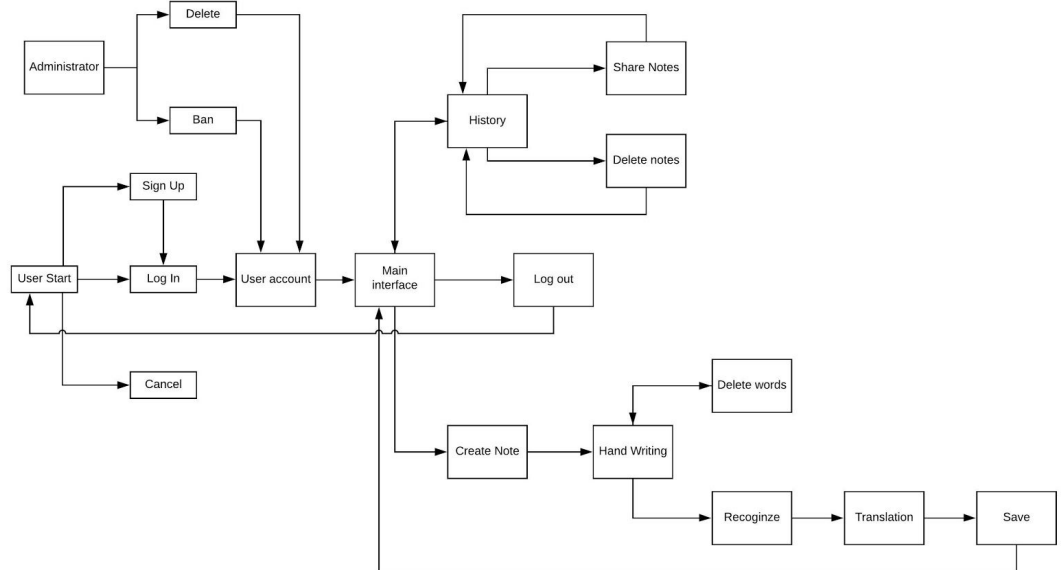
2. Administrators signIn the application it connect the login information from the user database to show the administrators' page. This page connected to the user database to show all the user accounts, and allow administrators to delete, ban and unban user accounts.



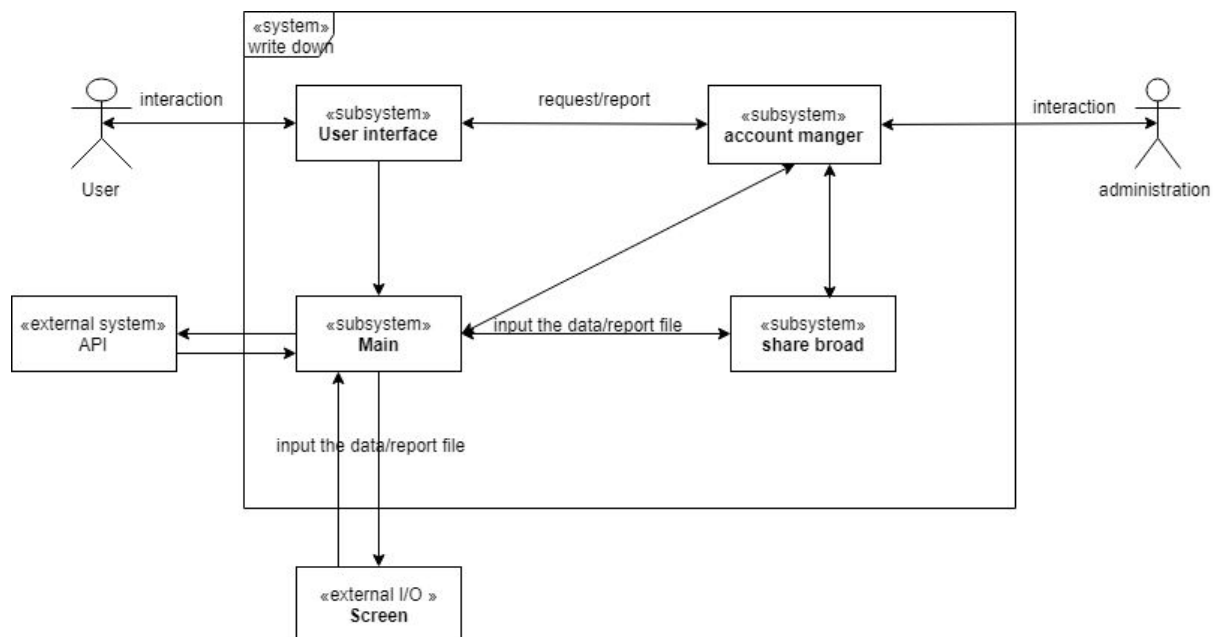
- Entity Relationship Model (E-R Model) (Yiheng Jiang)



- Overall operation - System Mode



- Simplified Sub-system to System interaction



5. System – Analysis Perspective – Group responsibility

- Identify subsystems – analysis point of view
- System (Tables and Description)
 - Data analysis
 - Data dictionary (Table - Name, Data Type, Description)(Yangling Hu)

Column Name	Data Type	Description
Uuid	VARCHAR()	Unique Id Key with random value, PK, UN
LastName	VARCHAR()	User lastname
FirstName	VARCHAR()	User firstname
Gender	INT()	0- For male 1- For female
Address	VARCHAR()	(optional)
Username	VARCHAR()	Input username with characters ,symbols and interages,UN
Password	VARCHAR()	Input password with characters, symbols and interages
Usertype	INT()	0 for the normal user 1 for the administrator
Email	VARCHAR()	Input Email by correct email address, UN
Banned	INT()	1- for banning user account 0- for normal user account use
Img	VARCHAR()	Img update for user setting
History	VARCHAR()	Store data to the database from user account

■

- Process models
- Algorithm Analysis
 - Big - O analysis of overall System and Sub-Systems
 - API/OCR loading $O(\log(n))$
 - Database $O(n)$

○

6. Project Scrum Report - Group Responsibility

- Product Backlog (Table / Diagram)
- Sprint Backlog (Table / Diagram)
- Burndown Chart

7. Subsystems

7.1 Subsystem 1 – Name 1 - Individual responsibility

- Initial design and model
 - Illustrate with class, use-case, UML, sequence diagrams
 - Design choices
- Data dictionary
- If refined (changed over the course of project)
 - Reason for refinement (Pro versus Con)
 - Changes from initial model
 - Refined model analysis
 - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint - Link to Section 6)
- Coding
 - Approach (Functional, OOP)
 - Language
- User training
 - Training / User manual (needed for final report)
- Testing

7.2 Subsystem 2 – Name 2 - Individual responsibility

- Initial design and model
 - Illustrate with class, use-case, UML, sequence diagrams
 - Design choices
- Data dictionary
- If refined (changed over the course of project)
 - Reason for refinement (Pro versus Con)
 - Changes from initial model
 - Refined model analysis
 - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint - Link to Section 6)
- Coding
 - Approach (Functional, OOP)

- Language
- User training
 - Training / User manual (needed for final report)
- Testing

7.3 Subsystem 3 – Name 3 - *Individual responsibility*

- Initial design and model
 - Illustrate with class, use-case, UML, sequence diagrams
 - Design choices
- Data dictionary
- If refined (changed over the course of project)
 - Reason for refinement (Pro versus Con)
 - Changes from initial model
 - Refined model analysis
 - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint - [Link to Section 6](#))
- Coding
 - Approach (Functional, OOP)
 - Language
- User training
 - Training / User manual (needed for final report)
- Testing

7.4 Subsystem 4 – Name 4 - *Individual responsibility*

- Initial design and model
 - Illustrate with class, use-case, UML, sequence diagrams
 - Design choices
- Data dictionary
- If refined (changed over the course of project)
 - Reason for refinement (Pro versus Con)
 - Changes from initial model
 - Refined model analysis
 - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint - [Link to Section 6](#))
- Coding
 - Approach (Functional, OOP)
 - Language
- User training
 - Training / User manual (needed for final report)
- Testing

8. Complete System – *Group responsibility*

- Final software/hardware product
- Source code and user manual – screenshots as needed - Technical report
 - Github Link :<https://github.com/lxmshisg/WriteDown>
- Evaluation by client and instructor
- Team Member Descriptions