WordUp - User Guide

By: `Team T14-3` Since: `August 2019`

1. Introduction

WordUp (WU) is a desktop-based application for english language learners who prefer having a digital version of a personal vocabulary booklet (otherwise known as a word bank). More importantly, WU is great for on-the-go revision, which is not easily achieved with traditional hardcopy word banks. WU provides convenience and efficiency of adding words in a word bank, and allows users to manage their word collection faster with a Command Line Interface (CLI), as well as having a Graphical User Interface (GUI) for more interactive functions and for better

visualisation of content stored.

2. Quick Start

1. Ensure you have Java `11` or above installed in your Computer.

2. Download the latest `wordup.jar` here.

3. Copy the file to an ***empty folder*** you want to use as the home folder for your WordUp.

4. Ensure that you have `JavaFX` installed.

5. Double click on the Jar file.

2.1. Help if the jar file does not open

1. [Windows] Open command prompt by typing cmd in the start bar.

2. Enter the following command: `java --module-path [path to javafx lib folder]C:\Java\javafx-sdk-11.0.2\lib --add-modules=javafx.controls -jar [path to home folder of WordUp]C:\home\wordup-v1.3.jar`

3. If you encounter a `Module Not Found` error, please check that you do not have a spacing in your file path to `javafx\lib`.

4. Type the command in the command box and press Enter to execute it. e.g. typing `help` and pressing `Enter` opens up the window showing a list of commands available.

2.2. Some example commands you can try

- `list` : shows the list of words currently stored in the Word Bank in alphabetical displayOrder or otherwise customised in the app settings

- `add``w/life m/the existence of an individual human being or animal` : adds a word tagged by "w/" with its corresponding meaning tagged "m/" into the Word Bank

- `delete``w/freedom` : deletes the Word Bank entry of the word "freedom"

- `exit` : exits the application

  1. Refer to Section 3, "Features" for in-depth details for more commands.

3. Features

> **Command Format**
>
> **Caution**: The tags and synonyms are currently not written to the storage file due to some issues on our side, so we have disabled the writing, all tags and synonyms will be cleared when the program is closed. We will fix this issue by Week 12 after we re-enable the saving
>
> - Words in `UPPER_CASE` are parameter values to be supplied by the user e.g. in `add w/WORD m/MEANING`, `WORD` and `MEANING` are parameters to be replaced by user input, such as in the form of `w/life` and `m/the existence of an individual human being or animal`.
>
> - Items in square brackets are optional e.g. `r/REMINDERDATE [TIME]` can be used as `r/24/02/2300 0900` or `r/24/02/2300`.
>
> - Items with ⋯ after them can be used multiple times including zero times e.g. `[t/TAG]`⋯ can be used as `t/negative, t/emotions, t/lesson3`

3.1. Viewing help : `help`

Shows available commands and their usage.
Format: `help [COMMAND]`

Examples:

- `help` shows all available commands.
- `help delete` shows the description and usage of delete command.

Welcome, what would you like to do today?

help delete

Delete a word or tag from wordbank.
Format: delete w/WORD_TO_BE_DELETED [t/TAG]

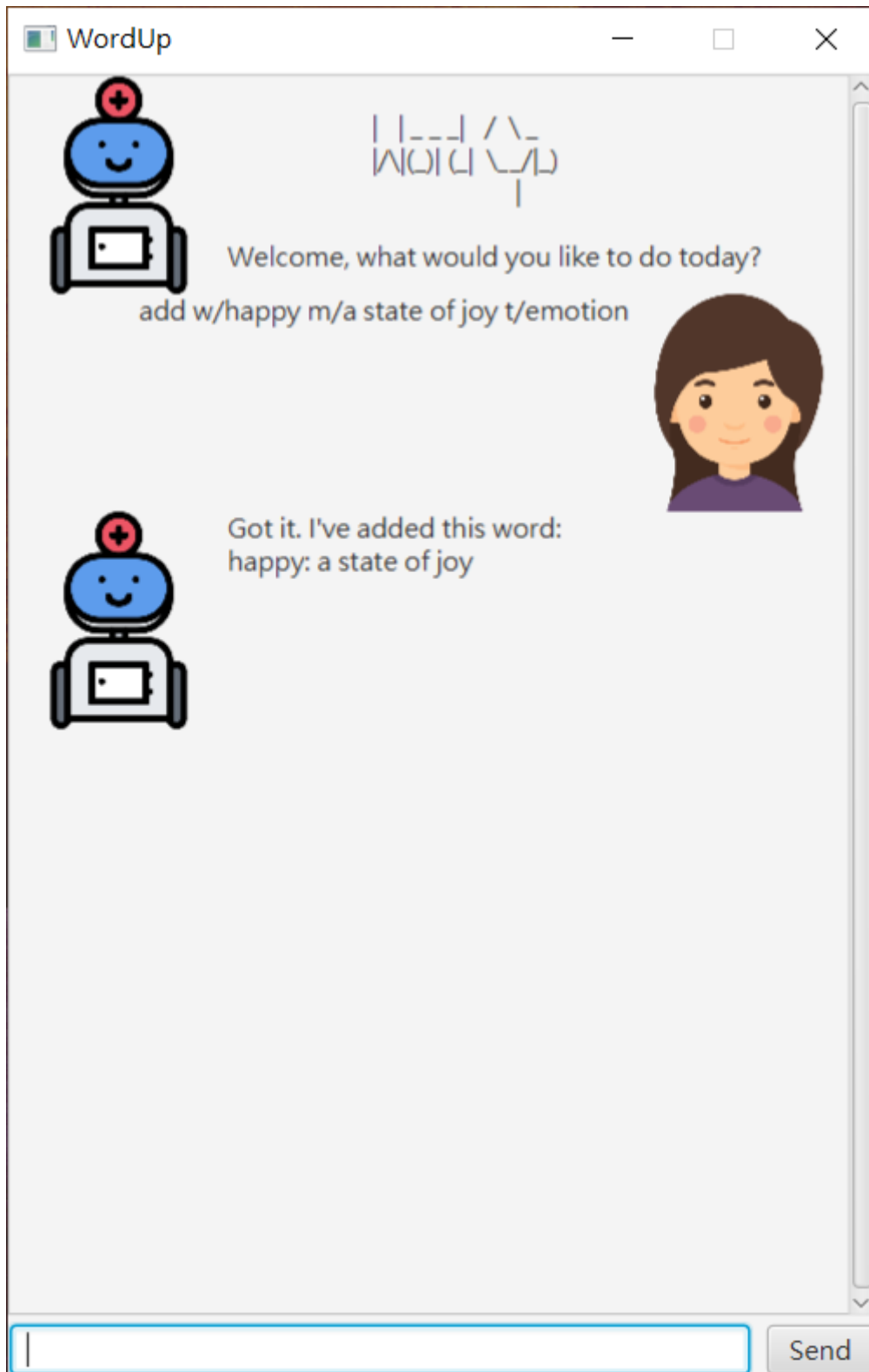## 3.2. Adding a word: add

Adds a word to the Word Bank.

Format: add w/WORD w/MEANING [t/TAG]···

Examples:

- add w/life m/the existence of an individual human being or animal t/meaningoflife

`t/livingit`

- `add w/happy m/a state of joy t/emotion`

3.3. Listing all words inside dictionary : `list`

Shows a list of words categorised as specified.
Format: `list [o/ORDER]`

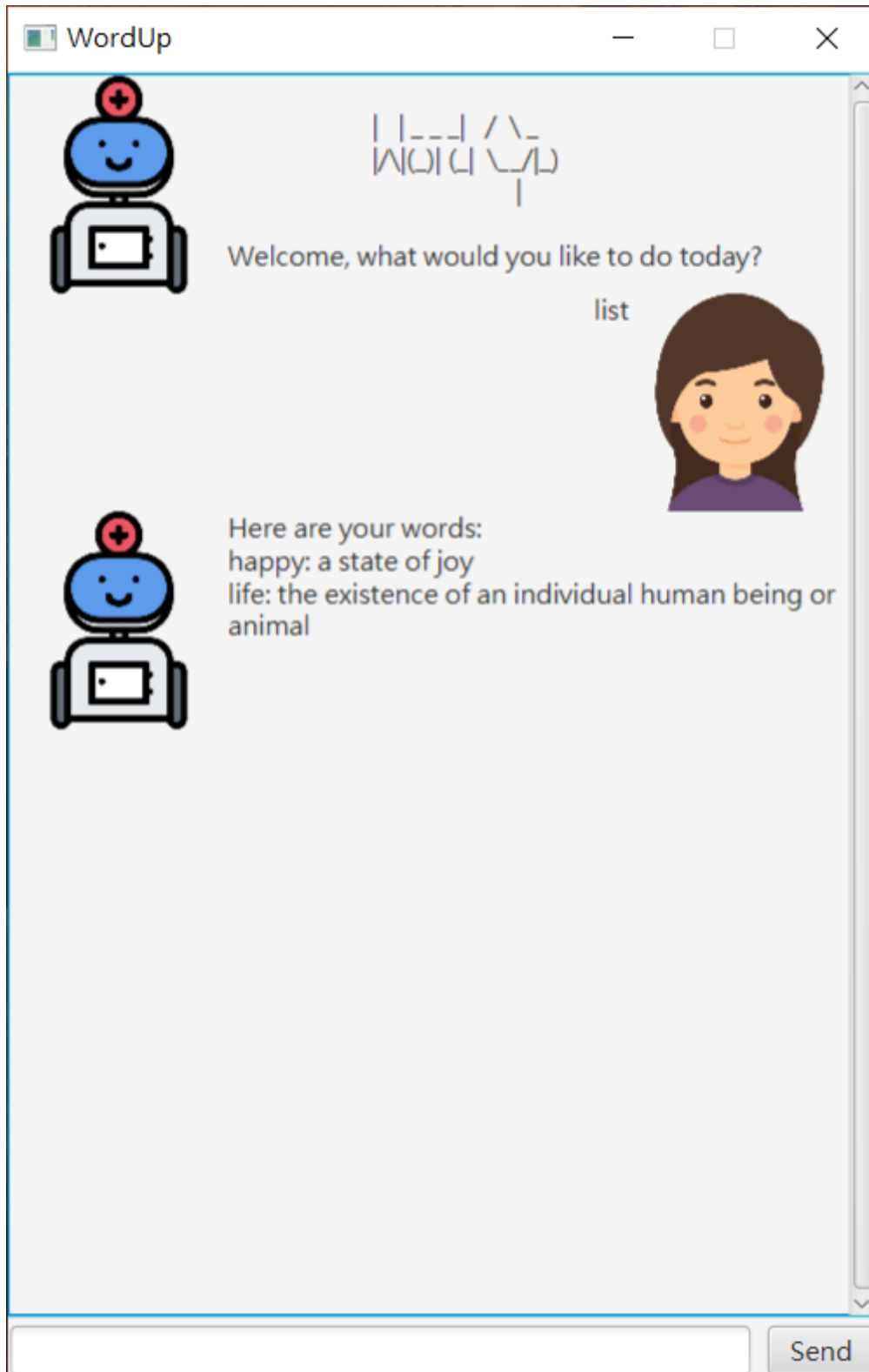Example: `list o/asc` shows the list of words tagged "emotions" in alphabetical displayOrder

List of possible o/ORDER values:

`asc` - displayOrder the list in ascending alphabetical displayOrder

`desc` - displayOrder the list in descending alphabetical displayOrder

3.4. Editing the meaning of a word : `edit`

Edits a meaning of a word currently present in the dictionary.

Format: `edit w/WORD m/NEW_MEANING`

• Both the WORD and NEW_MEANING must be provided for the edit to be successful.

Examples:

- `edit w/drink m/a liquid for consumption`
  Edits the word 'drink' and updates the meaning to "liquid for consumption".

WordUp — □ ✕

```
| |___| /\_
\\()| | \_/|)
       |
```

Welcome, what would you like to do today?

edit w/drink m/a liquid for consumption

Got it. I've edited this word:
drink: a liquid for consumption

Send

3.5. Searching for the meaning of words: `search`

Search for the meaning of words that is present in the dictionary. If word is not present, it will try to search the internet.

Please note that this search feature works on the singular form of words only. (E.g 'search w/potato' instead of 'search/potatoes')

Format: `search w/WORD`

- The search is case insensitive as they will be converted to lowercase. e.g `HaNs` will match `hans`

- Full words will be matched and given a result immediately.

- If word is similar to a record in the dictionary, a suggestion of similar words will be prompted.

Examples:

- `search w/drink`
  Returns `a liquid for consumption`

- `search w/potatoes`
  Returns suggestion of `potato`
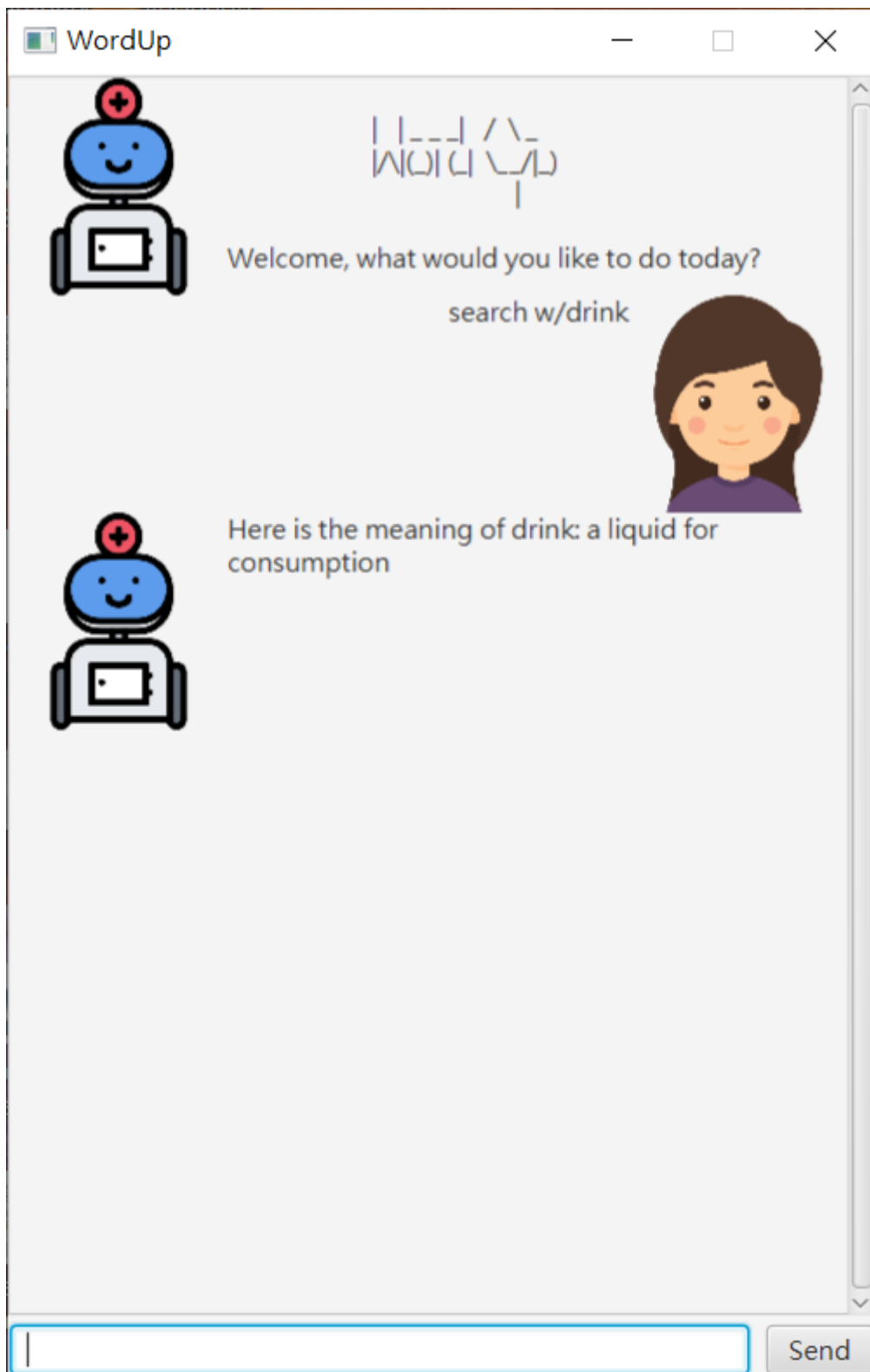
3.6. Checking search frequency: `freq`

Displays the list of words in ascending or descending order of search count.

Examples:

- `freq o/asc`
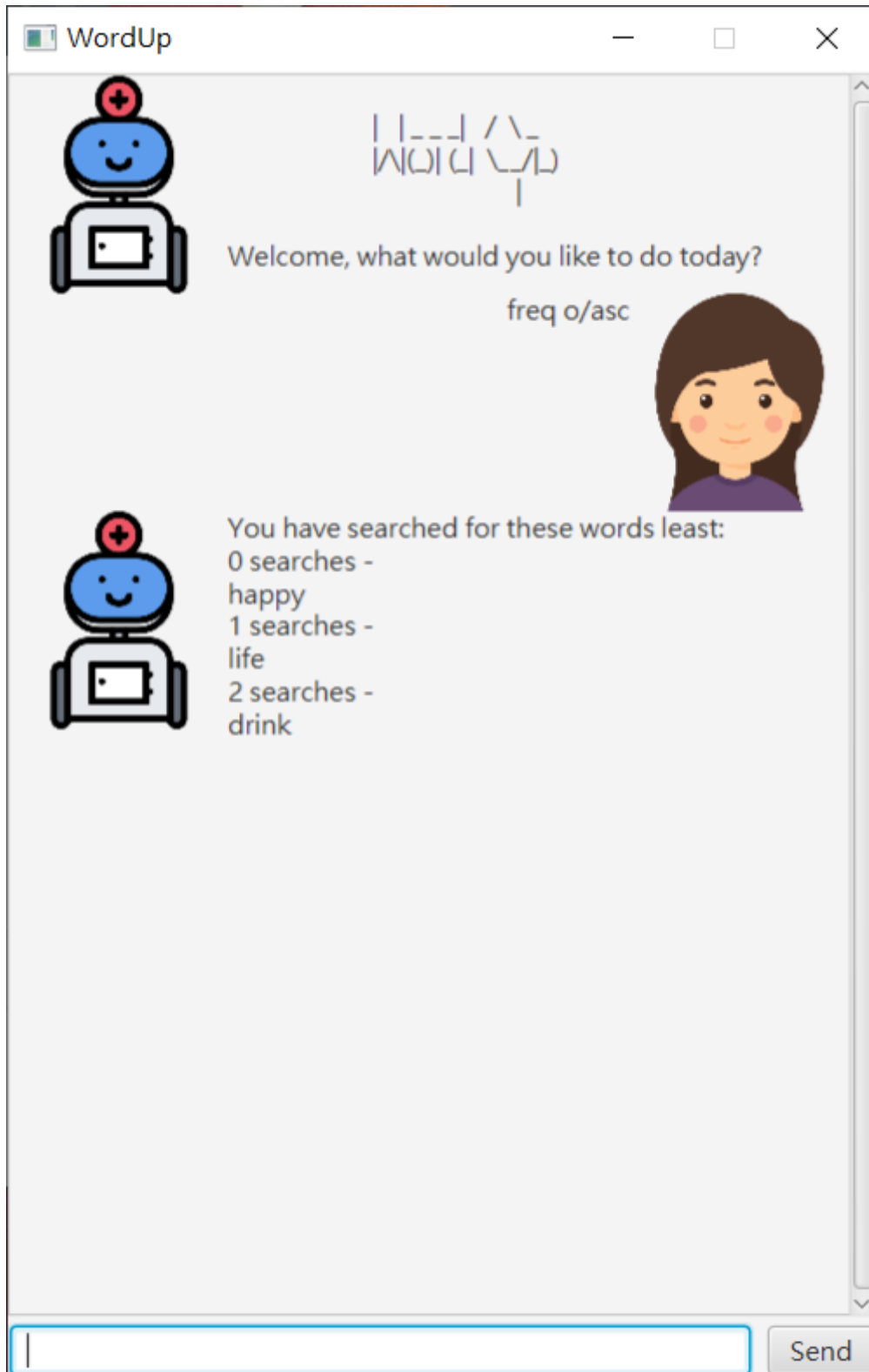 Returns a list of words in ascending order of search count (from lowest to highest search count).

List of possible o/ORDER values:

asc - displayOrder the list in ascending alphabetical displayOrder

desc - displayOrder the list in descending alphabetical displayOrder

## 3.7. Searching for words with their begin: `search_begin`

Search for the list words that in the dictionary that have a specific part.

Format: `search_begin w/WORD`

- The search is case insensitive as they will be converted to lowercase. e.g `HaNs` will match `hans`

- Full words will be matched and given a result immediately.

Examples:

- `search w/d`

- `search_begin w/d`
  Returns `drink, dog, do, etc`
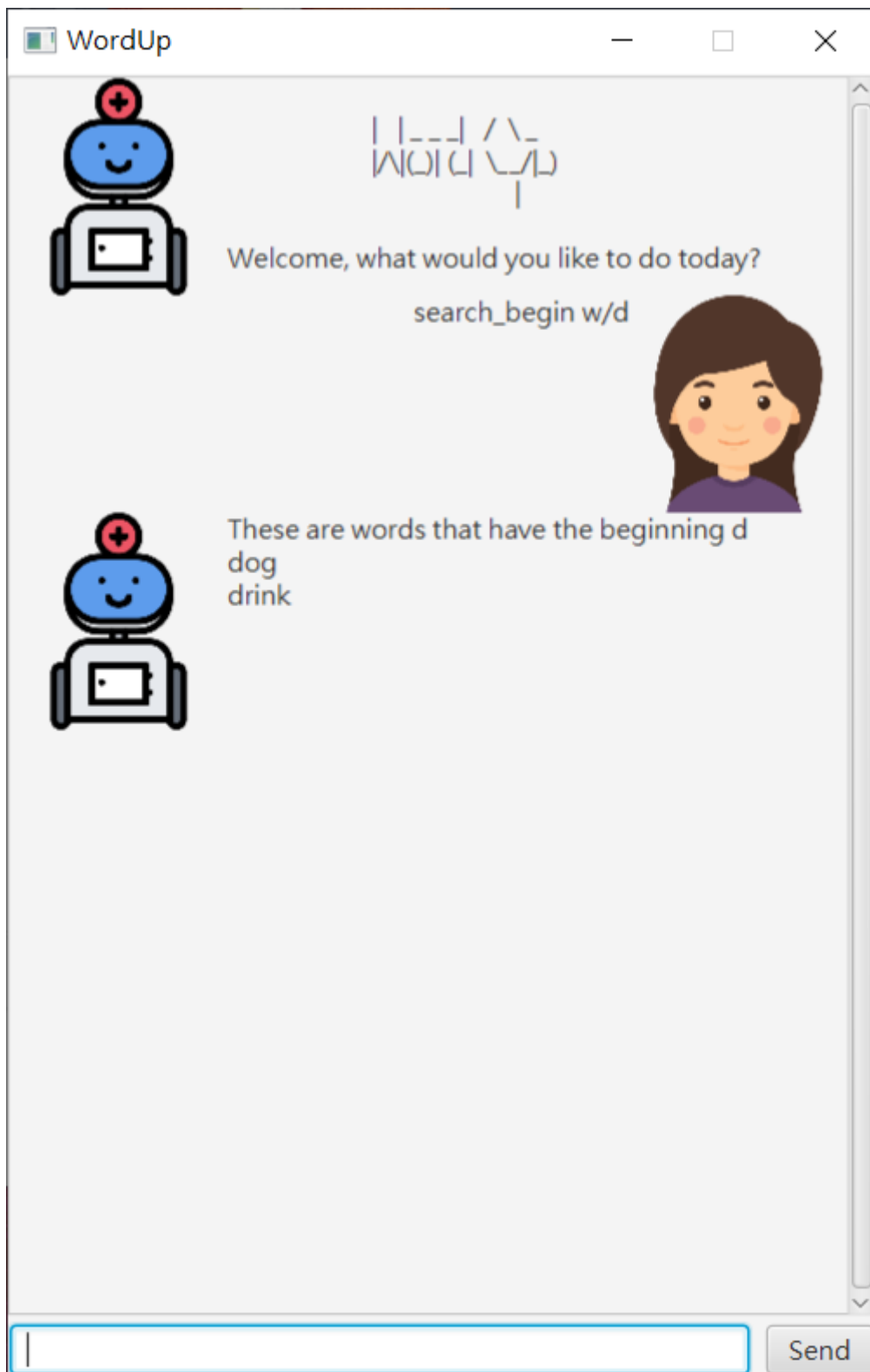
3.8. Generating quizzes: `quiz`

Generate quizzes from the wordBank to test the user's understanding.

Format: Type `quiz` to enter quiz mode.
In quiz mode, type `start` to start the quiz.
Type `exit_quiz` to exit the quiz.
When answering the MCQs, type the index 1, 2, 3, 4 to select the correct option.

In a single question WordUp will display a word, and user shall choose the corresponding meaning. A quiz includes 4 multiple choice questions.
At the end of a quiz, it will show user the score of the quiz and remind user the meaning of words that are wrongly answered.

Example:

```
What is the meaning of WordUp?
1. ...
2. a fantastic personal digital booklet
3. ...
4. ...
```

The correct answer is 2. Type 2 and WordUp will respond the correctness.

Let's do some quiz to enhance your word knowledge
Type "start" to begin quiz or "exit_quiz" to go back

start

What is the meaning of life?
1.a state of joy
2.the existence of an individual human being or animal
3.animal
4.a liquid for consumption

2

Yes!! The correct answer is "the existence of an individual human being or animal".

What is the meaning of happy?
1.the existence of an individual human being or animal
2.a state of joy
3.a liquid for consumption
4.animal

3.9. Deleting a word object : `delete`

Deletes the word object (inclusive of all meaning, synonyms and tags) from dictionary.
Format: `delete w/WORD [t/TAG]`

- Deletes the WORD object directly from the program and storage if no tags are included.

- If one or more tags are included in the command, only the tags will be deleted from the word.

Examples:

- `delete w/drink`
  Deletes the drink object from dictionary.
- `delete w/drink t/action`
  Deletes the tag action from word drink.
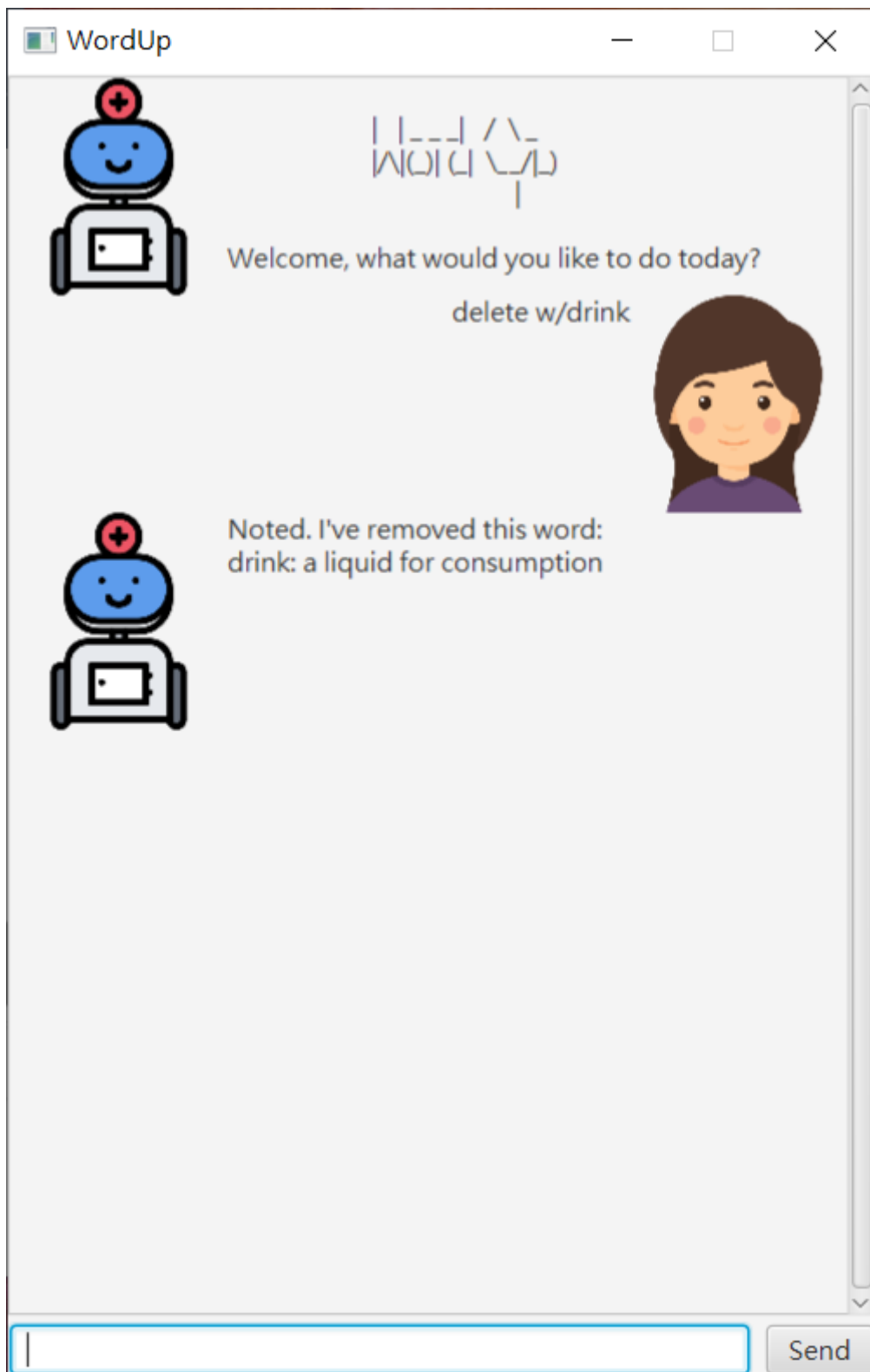
3.10. [Not supported] Clearing all entries : `clear`

Clears all entries from the dictionary.
Format: `clear`

[clear]

3.11. Scheduling words to study : `schedule`

Schedules reminders for a selected list of words given a user specified date and time.
Format: `schedule` OR `schedule w/word1, word2, ⋯ r/dd-MM-yyyy HHmm`

> - Users can set a reminder for multiple words at one go. Terminate word input by entering blank line.

Examples:

- `schedule w/happy, elated r/12-02-2020 1400`

OR

- Enter `schedule`
  Enter a list of 'enter key' separated words: `happy[Enter] elated[Enter]`
  Enter a date and time for the reminder: `12-02-2020 1400`

Reminder is then set. Summary of reminder details is shown. Reminder pop up will be shown on 12 Feb 2020 2pm.

NOTE	Due to the current implementation, you will not be able to exit the app while doing the multi-line setup of the reminder.
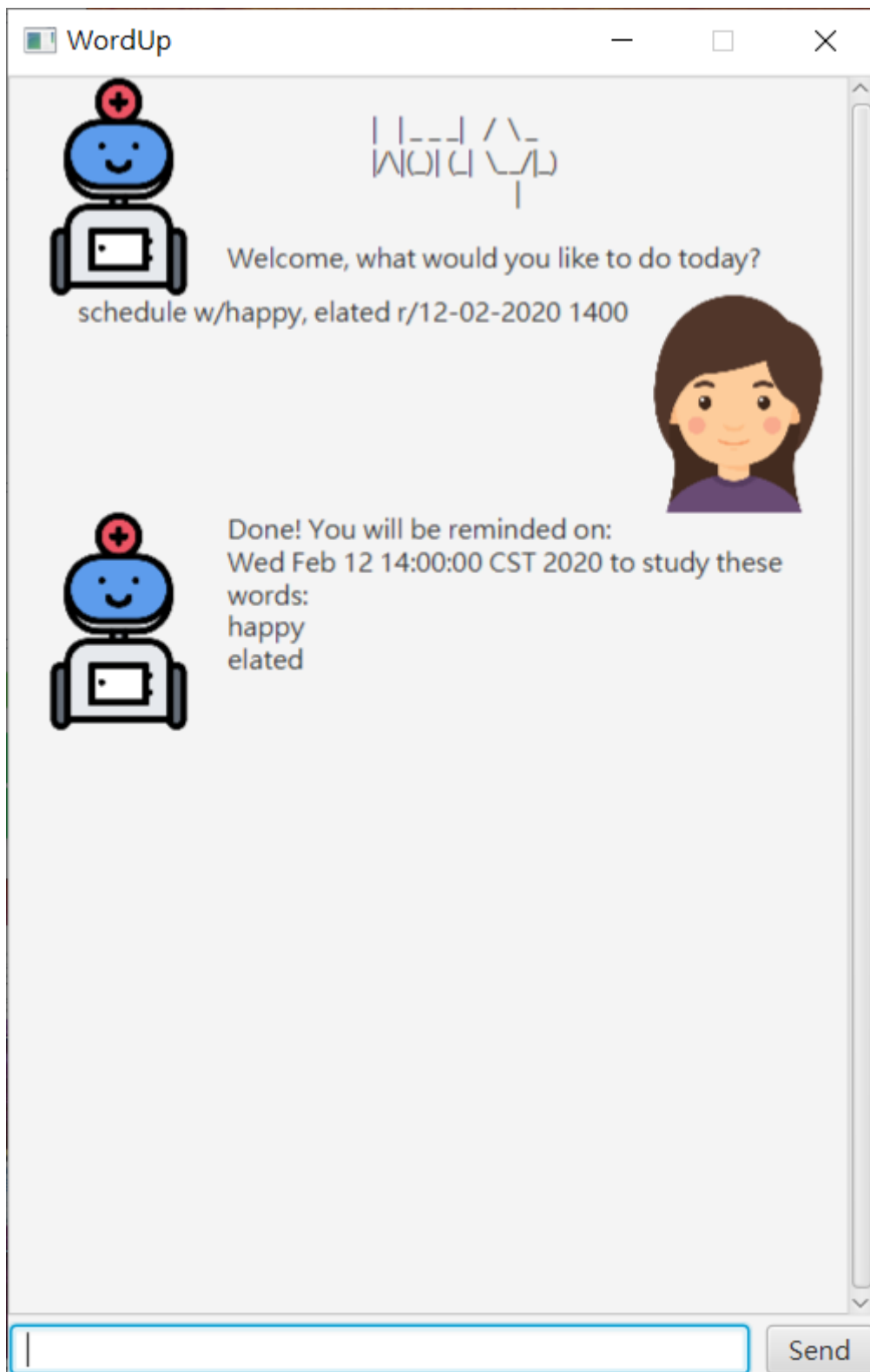
3.12. Checking recently added words : `history`

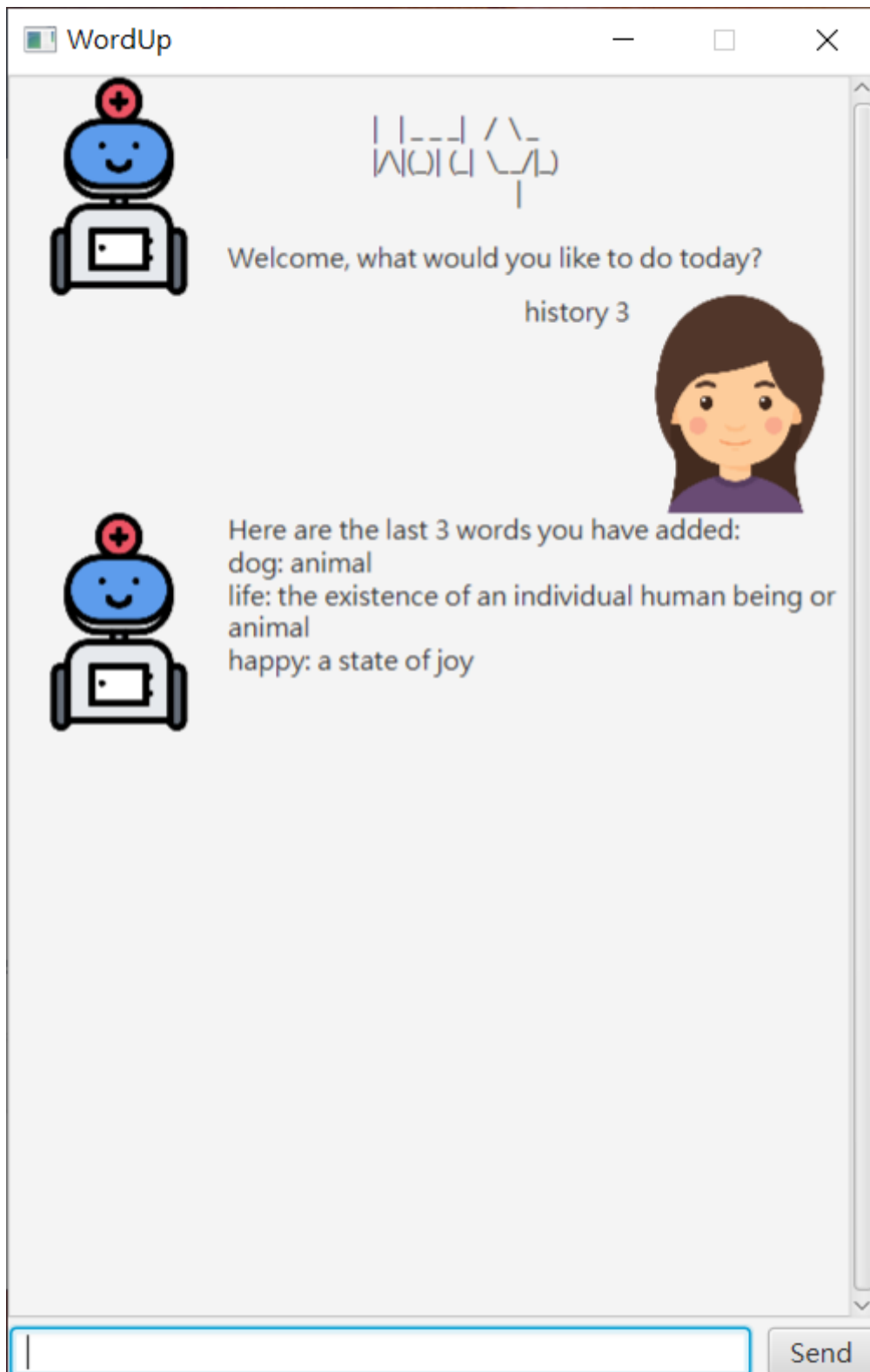Shows a list of recently added words by the user.

Format: `history INTEGER(number of past entries to be shown)`

- Entering a number larger than the number of words in the word bank returns all the words in the word bank.
- List is shown with the latest added words at the top.

Examples:

- `history 5`
  Shows the list of the last 5 words added, with the latest entry first.

3.13. Adding synonyms of a word : `addsyn`

Adds one or many synonyms to an existing word in the database.
Format: `addsyn w/WORD_TO_BE_TAGGED s/SYNONYM1 SYNONYM2`

If there are more than one synonym to be added, please separate them by spaces.

3.14. Searching synonyms of a word : `search_syn`

Search for al the synonyms to an existing word in the database.
Format: `search_syn w/WORD`

Synonyms are chained together, so even new words will be recognize.

3.15. Adding tags to a word : `tag`

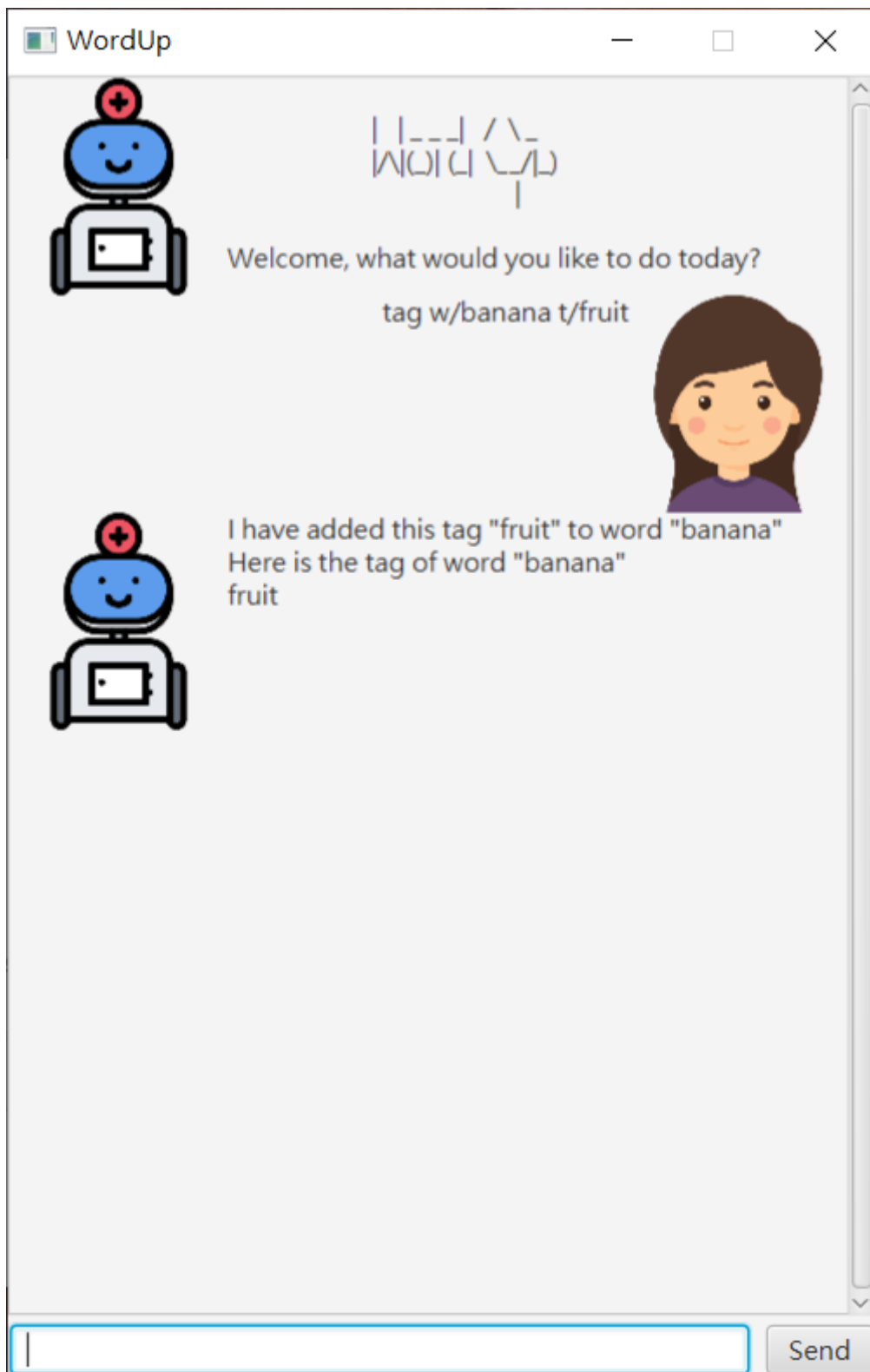Assigns a tag to an existing word in the database.
Format: `tag w/WORD_TO_BE_TAGGED [t/NEW_TAG]`

If there are more than one tag to be added, please use `t/` for each tag

Examples:

- `tag w/banana t/fruit`
  Assigns the word banana with tag fruit.

3.16. Searching for tags of a word or words of a tag: `search_tag`

There are 2 ways of using it, one is to search for all tags of a specific word, one is to search for add words tagged by a tag. If users want to search for tags of a word:

Format: `search_tag w/WORD`

Examples:

- `search_tag w/happy`

  Output: emotions, adjective, etc.

  This command lists out all tags of word `happy`.

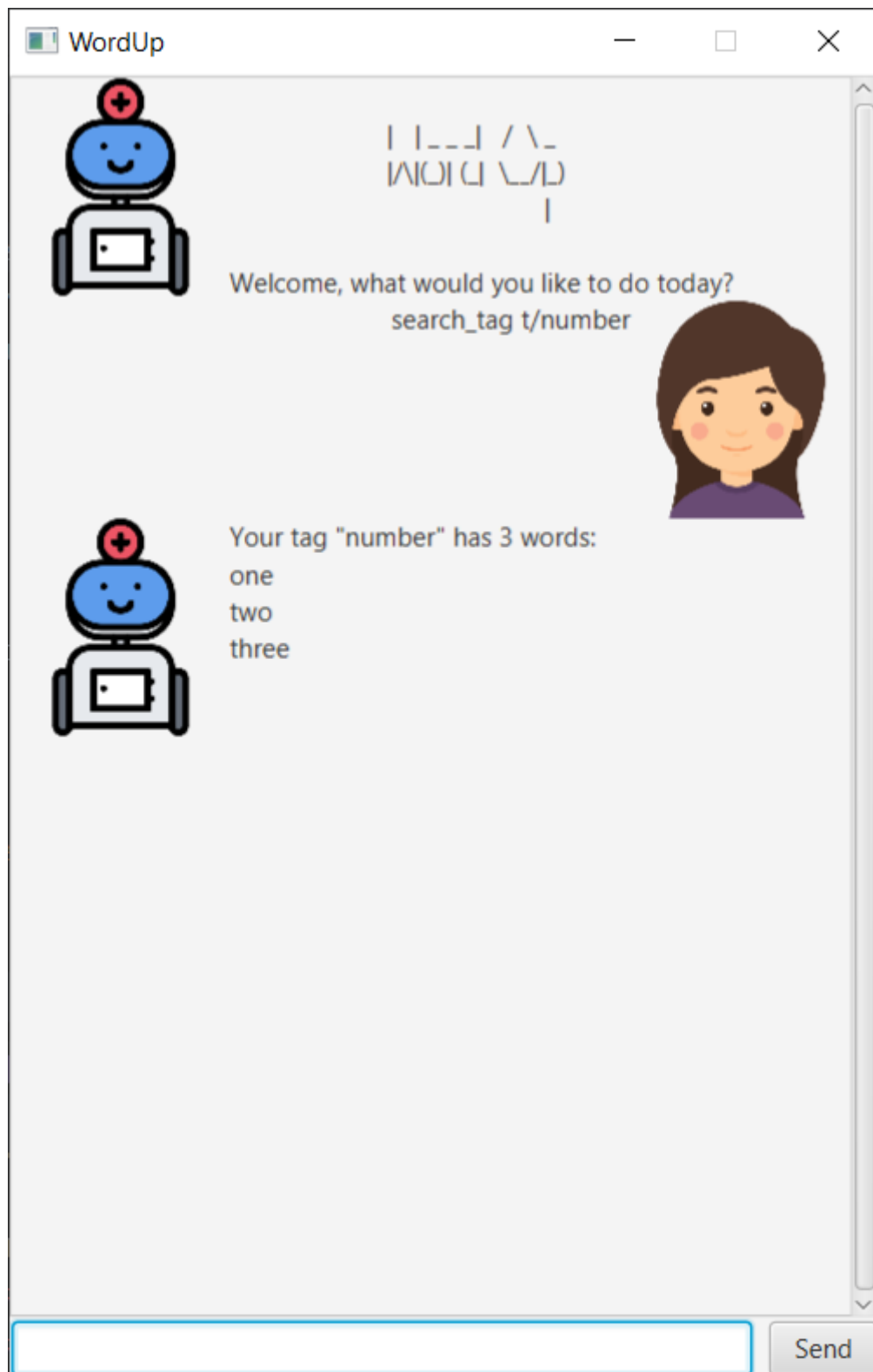If users want to search for all words of a tag:

Format: `search_tag t/TAG`

Examples:

- `search_tag t/numbers`

Output: one, two, three, etc.

This command lists out all words that are tagged by tag `numbers`

3.17. Listing all tags inside dictionary : `list_tags`

Shows a list of all tags that users have input

Format: `list_tags`

3.18. Adding example sentence : `add_example`

Add example sentence for a word. Format: `add_example w/ORDER e/EXAMPLE`

When user `search` a word, WordUp will display the example sentence along with the meaning if the example sentence exists.

3.19. Exiting the program : `exit`

Exits the program.
Format: `exit`

3.20. Saving the data

Dictionary data are saved in the hard disk automatically after any command that changes the data. There is no need to save manually.

3.21. Full integration into Excel/CSV `[coming in v2.0]`

{}

4. FAQ

Q: How do I transfer my data to another Computer?
A: Install the app in the other computer and place the data folder from the previous computer into the other Computer.

5. Command Summary

- **Add** `add w/WORD m/MEANING [t/TAG]⋯`
  e.g. `add w/life m/the existence of an individual human being or animal.`
- **Add Synonym** : `addsyn w/WORD s/SYNONYM1 SYNONYM2 SYNONYM3 ⋯`
  e.g. `addsyn w/water s/liquid beverage drink`
- **Add Example** : `add_example w/WORD e/EXAMPLE`
- **Delete** : `delete w/word`
  e.g. `delete w/life`
- **Delete tag** : `delete w/WORD t/TAG`
  e.g. `delete w/one t/number`
- **Edit** : `edit w/WORD m/NEW_MEANGING`
  e.g. `edit w/life m/the existence of a living thing.`
- **Exit** : `exit`
- **Frequency** : `freq [o/ORDER]`
  e.g. `freq o/asc`

- **Help** : `help [COMMAND]` e.g. `help add`

- **History** : `history INTEGER`
  e.g. `history 5`

- **List** : `list`

- **List Tags** : `list_tags`

- **Schedule** : `schedule → WORD ⋯ [Enter blank line] → DATE TIME`
  e.g. `schedule → happy[Enter] unhappy[Enter] bipolar[Enter][Enter] → 29-10-2019 2359`

- **Search** : `search w/WORD`
  e.g. `search w/life`

- **Search Prefix** : `search_begin w/WORD`
  e.g. `search_begin w/a`

- **Tag** : `tag w/WORD t/TAG1 t/TAG2 t/TAG3`
  e.g. `tag w/happy t/emotion t/delight`

- **Search tag** :
  `search_tag w/WORD`
  e.g. `search_tag w/one`
  `search_tag t/TAG`
  e.g `search_tag t/number`\`

- **List tags** : `list_tags`

- **History** : `history INTEGER`
  e.g. `history 5`

- **Schedule** : `schedule → WORD ⋯ [Enter blank line] → DATE TIME`
  e.g. `schedule → happy[Enter] unhappy[Enter] bipolar[Enter][Enter] → 29-10-2019 2359`

- **Search Frequency** : `freq o/ORDER`
  e.g. `freq o/asc`

- **Exit** : `exit`

- **Search Synonym** : `search_syn w/WORD`
  e.g. `search_syn w/life`

- **Quiz** : `quiz → start`