

WANG, Zeyu's notes

1. Characteristics:

Realtime Input Checking Logic

I developed an automatic input checking function using boundary class (InputChecker classes in my source code) for my tasks. If the user input is invalid, for example, the input year is not between 1880 and 2019, it will disable the generate button and show a corresponding warning message next to the invalid input. This will guarantee that no input can actually go into the system and cause exception or crash the software.

Here are some examples

The screenshot shows a web application window titled "Team T-29: Popular Names". It has a navigation bar with tabs: "Task Zero", "Reporting 1", "Reporting 2" (selected), "Reporting 3", "Application1", "Application 2", and "Application 3". The main content area has a box titled "K-th Popular Names" with the text "A report on k'th popular name during start year - end year will be generated." Below this are input fields for "Start Year:" (2019), "End Year:" (1980), "Rank:" (5), and "Gender:" (radio buttons for Male and Female, with Female selected). To the right of the "Start Year" and "End Year" fields, there are red error messages: "Start year should be an integer within 1880 to end year" and "End year should be an integer within start year to 2019". At the bottom right are "Generate" and "Clear" buttons. A "Console" tab is visible at the bottom left.

The screenshot shows the same web application window. The "Start Year" is now 1900 and the "End Year" is 1980. The "Rank" field now contains "10000", which is highlighted with a blue border. To the right of the "Rank" field, there is a red error message: "Rank too large. Some year does not have that many names". The "Gender" field remains the same. The "Generate" and "Clear" buttons are still at the bottom right. The "Console" tab is visible at the bottom left.

Team T-29: Popular Names

Task Zero

Reporting 1

Reporting 2

Reporting 3

Application1

Application 2

Application 3

K-th Popular Names

A report on k'th popular name during start year - end year will be generated.

Start Year:

1900

End Year:

2020

End year should be an integer within start year to 2019

Rank:

10000

Gender:

☐ Male

☒ Female

Generate

Clear

onsole

Team T-29: Popular Names

Task Zero

Reporting 1

Reporting 2

Reporting 3

Application1

Application 2

Application 3

Soul Mate Name Recommendation

Recommened soul mate names derived from two algorithms will be shown.

My Name

tommmm

Can not find this male name in year 1960

My Birth Year

1960

My Gender

☒ Male

☐ Female

Mate Gender

☐ Male

☐ Female

Age Preference

☐ Younger

☐ Older

Max Age Difference

Generate

Clear

Team T-29: Popular Names

Task Zero Reporting 1 Reporting 2 Reporting 3 Application1 Application 2 Application 3

Soul Mate Name Recommendation

Recommened soul mate names derived from two algorithms will be shown.

My Name Please input a name with English characters only

My Birth Year Please input an 4 digits integer

My Gender ☐ Male ☒ Female

Mate Gender ☐ Male ☐ Female

Age Preference ☐ Younger ☐ Older

Max Age Difference

Generate Clear

2. Algorithm Analysis

My recommending algorithm runs as follows

- (1) First according to the user's age preference calculate the period to analyze (e.g., if user prefer a younger soulmate who is at most 10 years smaller than him, the period will be the user's birth year – 10)
- (2) For each year in the period, find the rank of user's name in that year, and then calculate a range (in current setting, $\text{range} = \text{myRank} / 100 + 2$) and discover all names in that year that is ranked within user's rank – range till user's rank + range.
- (3) Sort all the names discovered for this period by the total number of being discovered and the first one will be recommended because this name is "close" to the user's name.

The range need to be set appropriately because:

For names with high ranks, their ranks in recent years are more likely to be similar (e.g., Mary is always the top-ranked name in 1900's and Dorothy is always ranked around 3rd), thus the range needs to be smaller to find the best matching.

For names with low ranks, their ranks may be very distinct in consecutive years, thus the searching range needs to be larger to find a good matching.

Also the range should not be too large otherwise the program will run too slowly.