Task 1 (Reporting 1)

Brief description

Generate a brief report in response to the queries on the most popular names registered at birth over a given period.

Assumption

1. Assume the user can only be male or female

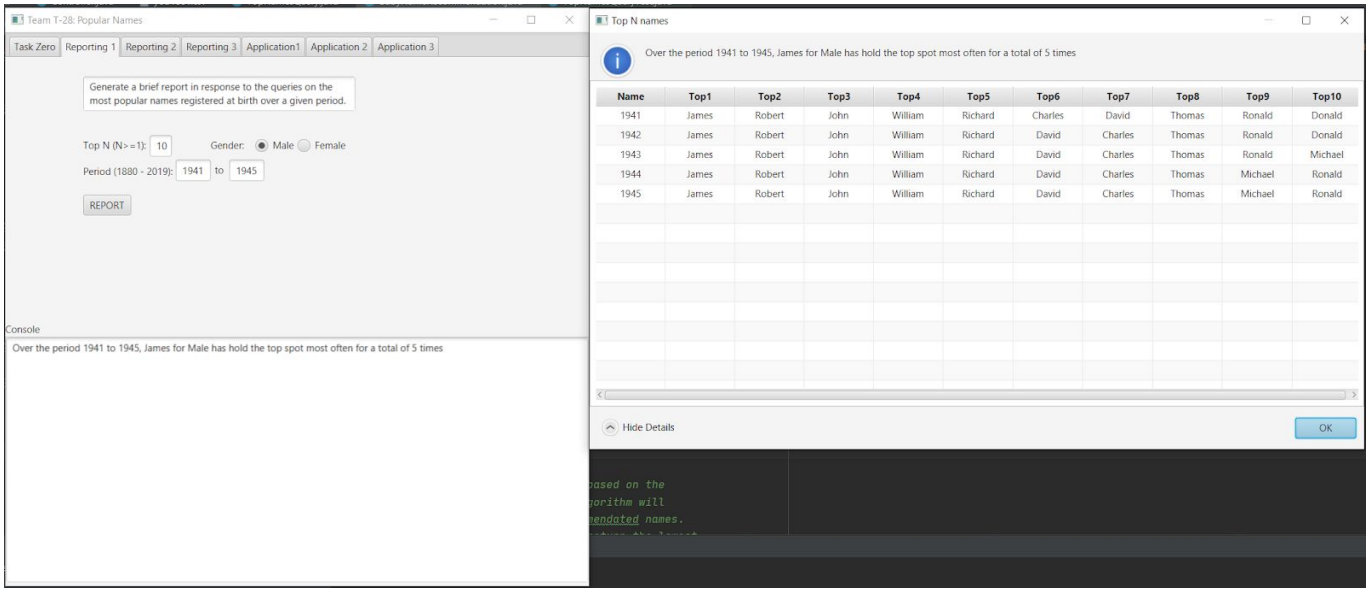
Limitation

1. Top N must be larger than or equal to 1
2. The period must be between 1941 and 1945
3. The start year must be earlier or equal to the end year

Inputs

1. Top N names wanted
2. Gender

Interpretation of Output



The Table

The first column states each year between the selected range
The second column to the N+1th column lists out the names of each rank, with the 2nd column having the top 1 name and the N+1th column having the Nth ranked name
Each row represents the record of a year .

The summary

The summary will state the name with the highest occurrence over the selected Period, if more than one name has the highest occurrence, all the names with the highest occurrence will still be listed out

Task 2 (Reporting 2)

Brief description

This task generates a report on the popularity of a particular name over a given period.

Assumption

1. Non-blank name (name with only whitespaces) is assumed to be a valid name, even names with blanks between letters are considered valid, as people can change their name to whatever weird name they like.
2. In a year, names with the same number of occurrences (count) will have the same rank.
3. Names are assumed to be case-sensitive, “David” is not equal to “david”.

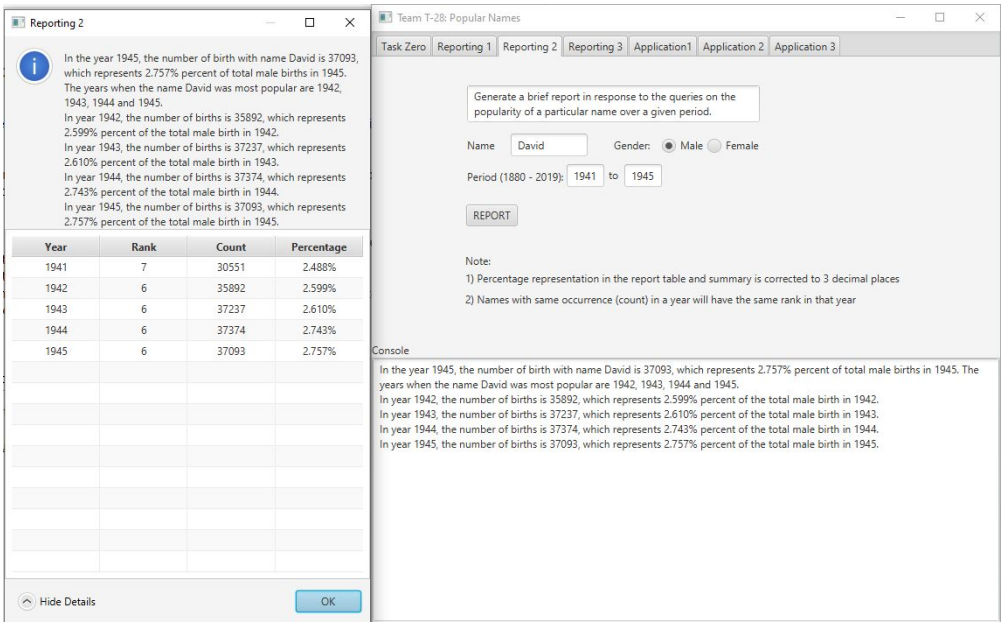
Limitation

1. The period to be queried must be between 1880 and 2019, as the data set only contains data from 1880 to 2019.
2. The name to be queried cannot be blank.
3. The gender of the user can only be “male” or “female”.
4. The percentage representation can only be shown to 3 decimal places due to the limitation of space in the GUI.

Inputs

Name: the name to be queried
Gender: gender of the name to be queried
Period: the period to be queried

Interpretation of Output



The table:

In each row, “Year” column indicates which year the data on the row corresponds to.
“Rank” column represents the rank of the name in that year.
“Count” column represents the number of occurrences of the name in that year.
“Percentage” column represents the ratio of the number of occurrences of the name to the total birth of the same gender

The summary (on both pop-up window & Console) consists of:

1) the popularity data of the name in the end year of the period
2) The year(s) that the name obtain its highest rank in the period (can be multiple years)
3) the popularity data of the name in each year it obtained its highest rank in the period

Task 3 (Reporting 3)

Brief description

This task generates a report on the trend of the top names over a given period.

Assumption

1. In a year, names with the same number of occurrences will have the same rank.
2. Names are assumed to be case-sensitive, “David” is not equal to “david”.

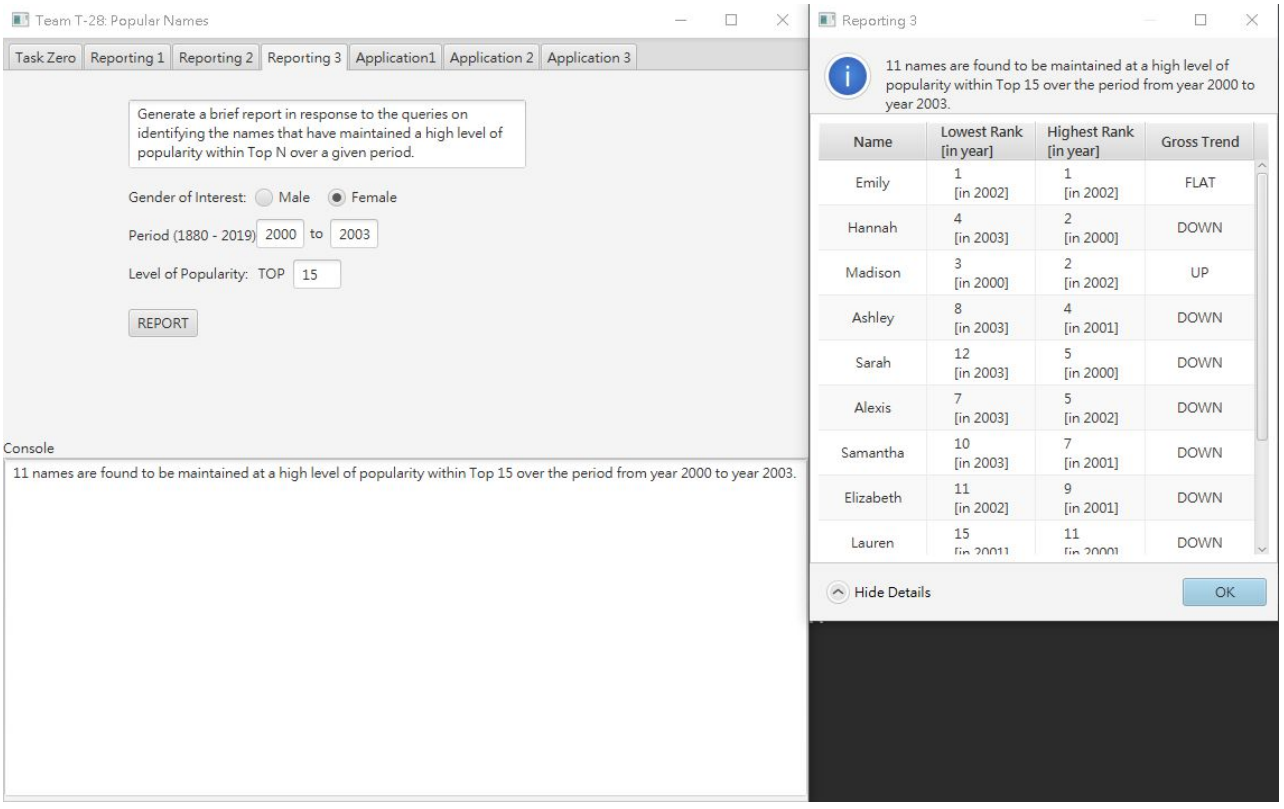
Limitation

1. The period to be queried must be between 1880 and 2019, as the data set only contains data from 1880 to 2019.
2. The gender of interest can only be “male” or “female”.
3. The query takes a long time to complete when either the queried period is long (difference between the years is large) or the level of popularity is high (N is large).

Inputs

Gender of Interest: gender of the name to be queried
Period: the period to be queried
Level of popularity: the number of top names to be queried

Interpretation of Output



The table:

For each row, the first three columns are the names, lowest rank with corresponding year and highest rank with corresponding year. The last column describes the trend, where “DOWN” indicates the year of the lowest rank is smaller than that of the highest rank, “UP” indicates the reverse, and “FLAT” indicates the years of both the lowest rank and the highest rank are identical.

The summary:

A brief summary showing the number of names that maintain in the top N within the given period.

Task 4 (Application 1)

Brief description

This task will provide 2 recommended names for a child, 1 for a boy 1 for a girl based on the ranking of the names of their parents. The system will generate 2 recommended names based on the ranking of the parents' names in their year of birth and find 2 names with the same ranks in the given vintage year.

Assumption

1. Assume the child must be male or female
2. Assume the parents must consists of 1 male (as the dad) and 1 female (as the mother)
3. Assume the year of birth of the parents must be later than 1880 and before 2019

Limitation

1. Vintage year must be between 1880 and 2019

Inputs

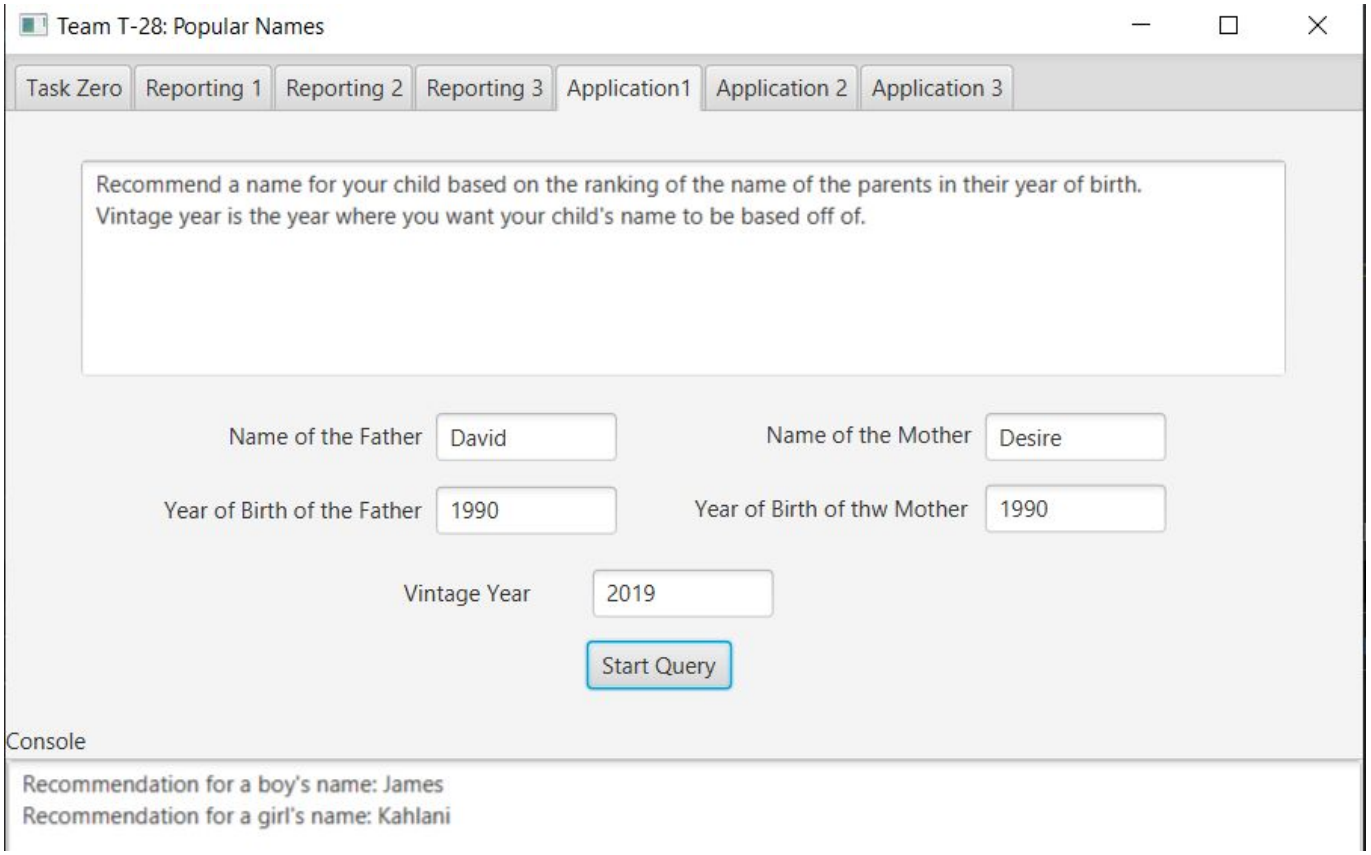
1. Name of the Father
2. Name of the Mother
3. Year of birth of the Father
4. Year of birth of the Mother
5. Vintage year

Algorithm

The algorithm returns 2 recommended names for the user, 1 for a boy, 1 for a girl based on the the ranks of the names of the parents in their respective birth years. Then the algorithm will choose 2 names with the same rank as the names of the parents and return the recommended names. If the rank of the parents' names are too low or out of bound, the algorithm will return the lowest ranked names in the vintage year

(Interpretation and Justification for Output on the next page)

Interpretation of the output



The console will show 1 recommended name for a boy and 1 recommended name for a girl

Justification for Output

The original NK-T4 algorithm of Universal compatibility will first compute the ranks of the parents, which is the rank of the parents' names' in their respective year of birth. Then it will compute a name for the boy and a name for the girl which has the same rank as the parents' names' ranks in their respective year of birth. However, the given algorithm will be unable to provide a recommendation when the ranks of the parents'
names are too low, or when the vintage year does not have a name at that rank.

Therefore, we modified the algorithm to make it return the name with the lowest rank in that year, if there are multiple names with the same rank, it will return the first name in that rank.

Task 5 (Application 2)

Brief description

This task makes predictions on names for compatible pairs (soulmate), which user can input his/her personal details and preference on the soulmate, and get a name prediction (recommendation) on his/her soulmate.

Assumption

1. Non-blank name (name with only whitespaces) is assumed to be a valid name, even names with blanks between letters are considered valid, as people can change their name to whatever weird name they like.
2. In a year, names with the same number of occurrences (count) will have the same rank.
3. Names are assumed to be case-sensitive, “David” is not equal to “david”.

Limitation

1. The year of birth of the user must be between 1880 and 2019, as the data set only contains data from 1880 to 2019.
2. The name of the user cannot be blank.
3. The gender of the user and the soulmate can only be “male” or “female”.
4. The preference of age of the soulmate can only be “younger” or “older” than the user.
5. Users with year of birth 1880 cannot choose “younger” soulmate, while users with year of birth 2019 cannot choose “older” soulmate, as the data set only contains data from 1880 to 2019.

Inputs

User:

Name:	name of the user seeking advice
Gender:	gender of the user seeking advice
Year of Birth:	year of birth of the user seeking advice

Details of Compatible Pair (Soulmate) to be matched:

Gender:	the user preference on gender of the compatible pair (soulmate)
Age preference:	the user’s preference of having a younger or older soulmate

Algorithm

The algorithm predicts the name of the soulmate of the user by finding the name that has the closest rank to the user's name rank with the preferred gender in the calculated soulmate's year of birth.

The algorithm first calculates the rank of the user's name in his/her year of birth.

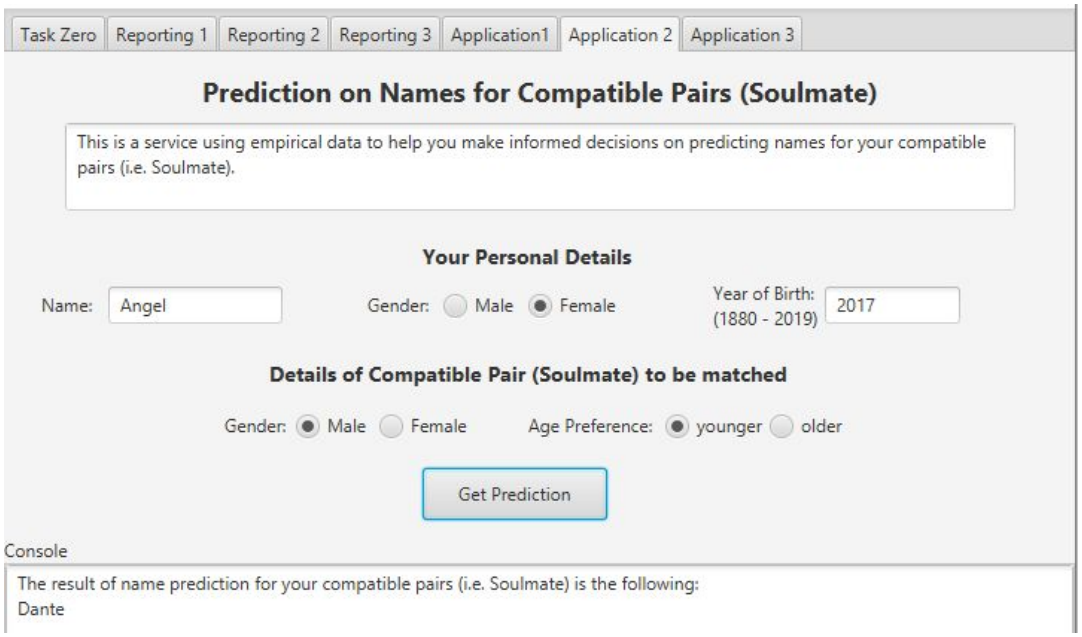
Then, calculate the year of birth of the soulmate, if the preference is "younger", the soulmate's year of birth is (user's year of birth - 1), else if the preference is "older", the soulmate's year of birth is (user's year of birth + 1).

Finally, the algorithm finds the name in the soulmate's year of birth that has the same rank as the user's name rank, which has the closest possible rank to the user's name rank, and returns that name.

If there do not exist that has the same rank as the user's name rank, either due to the user's name is not ranked, or the user's name rank is too low that there is not a name in the soulmate's year of birth with that rank, the algorithm returns lowest-ranked name in the soulmate's year of birth, which has the closest possible rank with name that is not ranked.

(Interpretation and Justification for Output on the next page)

Interpretation of Output



In the Console, this application will show the predicted name of the soulmate according to the user inputs.

Justification for Output (the recommendation)

This algorithm predicts the soulmate's name by finding a name in the soulmate's year of birth with the closest rank to the user's name rank in the user's year of birth.
This is because the rank of the name basically represents the popularity of a name in the year (as higher rank name has larger occurrence). With so many possible names, it is very difficult for the name of 2 people to be of the same rank and be equally popular in their year of birth.

The closer the ranks of the 2 names are, the lower the possibility is.
Such rarity resembles the concept of serendipity, which we use as the criteria for predicting the name of the soulmate of the user.

In addition, the soulmate's year of birth (user's year of birth - 1) or (user's year of birth + 1) because people in the same generation should have more common knowledge and interest, so age difference of 1 should make 2 people a good pair.

The original NK-T5 Algorithm of Universal Compatibility also uses the above principle to predicts the soulmate's name.
However, if the user's name is not ranked, or the user's name rank is too low that there is not a name in the soulmate's year of birth with that rank, the original algorithm uses the top-ranked name in the soulmate's year of birth as the predicted name. Thus, the predicted name will not have the closest rank to the user's name, but instead having the largest difference in rank and popularity. This does not resemble the concept of serendipity, and the name prediction is not coherent in all cases.

Therefore, the modified algorithm returns the lowest-ranked name in the soulmate's year of birth in such case. As a name that is not ranked is below the lowest-ranked name, the lowest-ranked name will have the closest rank to the not ranked name. Thus, the modified algorithm will always recommend the name with the closest rank to the user's name rank, making coherent recommendations in all cases.
Therefore, the modified algorithm gives justifiable and coherent predictions for the soulmate name in all cases.

Task 6 (Application 3)

Brief description

This task makes predictions on the compatibility of a pair, where the user can input his/her personal details and preference on the soulmate, including name, gender and age. A prediction on the compatibility between the user and the soulmate will be computed based on the given information.

Assumption

1. Non-blank names (names with only whitespaces) are assumed to be valid names and names with whitespaces between letters are considered valid, as people can change their name to whatever they like.
2. In a year, names with the same number of occurrences (count) will have the same rank.
3. Names are assumed to be case-sensitive, “David” is not equal to “david”.

Limitation

1. The year of birth of the user must be between 1880 and 2019, as the data set only contains data from 1880 to 2019.
2. The name of the user and the soulmate cannot be blank.
3. The gender of the user or the soulmate can only be “male” or “female”.
4. The preference of age of the soulmate can only be “younger” or “older” than the user.

Inputs

User:

Name:	name of the user
Gender:	gender of the user
Year of Birth:	year of birth of the user

Details of soulmate to be matched:

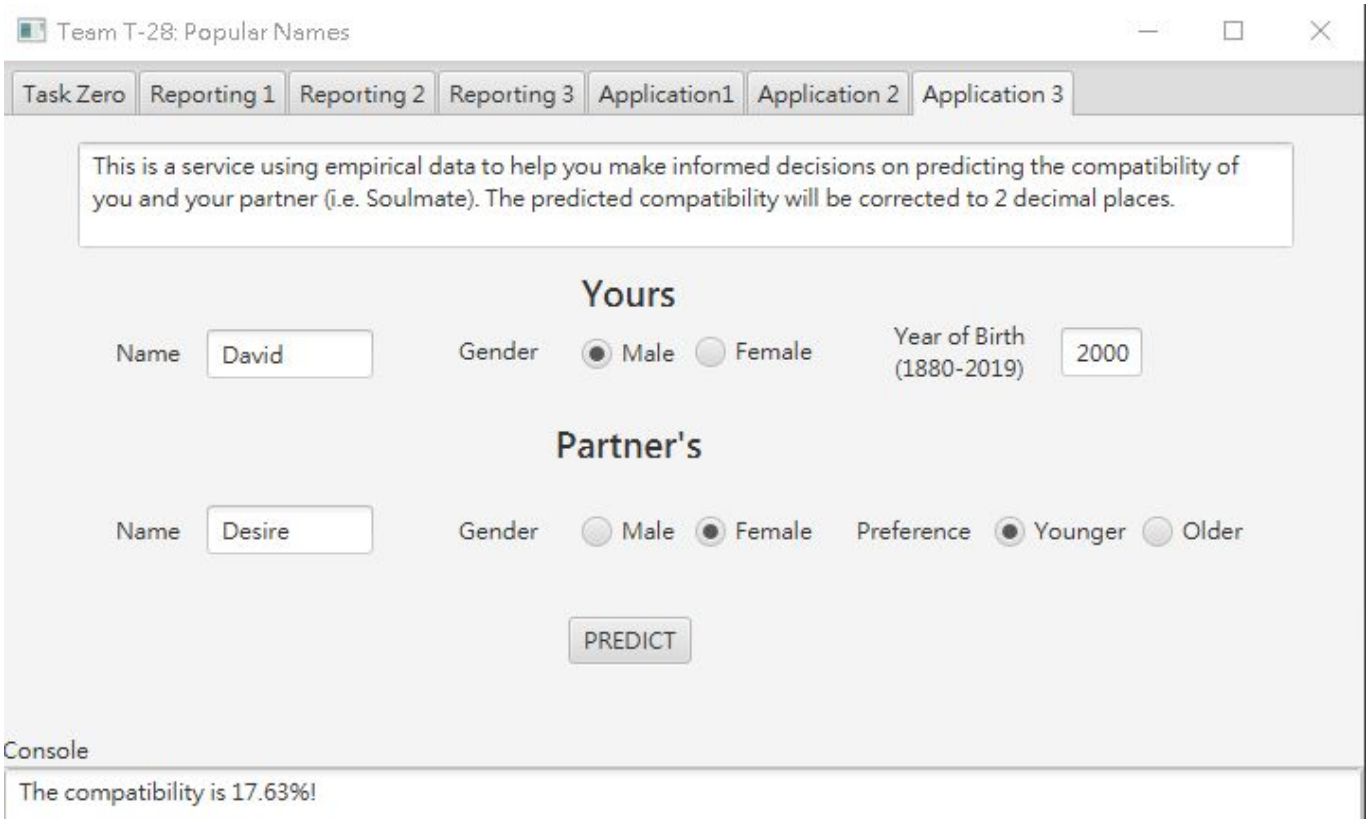
Name:	the user’s preference on the name of the soulmate
Gender:	the user’s preference on the gender of the soulmate
Age preference:	the user’s preference on the age of the soulmate

Algorithm

The algorithm computes the difference between the ranks of the names of the pair in their own year, and the maximum possible difference between the ranks of the names of the pair in their own year.
The ratio implies how close their ranks are and is then converted to percentage (corrected to 2 d.p.).
The percentage is the predicted compatibility of the pair.
Exception case is when any of the names of the pair is not ranked on the corresponding year, the algorithm will substitute it with the lowest possible rank of the corresponding gender of the corresponding year.

(Interpretation and Justification for Output on the next page)

Interpretation of Output



In the console, the predicted compatibility is shown.

Justification for Output

The original NK-T6 Algorithm of Universal Compatibility computes a variation of the percentage difference, which will result in values that are smaller than 0% or greater than 100%.

The modified algorithm uses the rank difference as the numerator and the maximum difference as the denominator.

Since the difference is always greater than or equal to zero, and is always smaller than or equal to the maximum difference, the result will always be between 0% and 100%, which is consistent with the expected output format.

The algorithm is used because we believe it is difficult for the two names of the pair to have the same rank. The closer the ranks are, the lower the possibility is. Such rarity resembles the concept of serendipity, which we use as the criteria for predicting compatibility.

The reason why the algorithm substitutes the rank of names that is not ranked with the lowest rank of the corresponding gender of the corresponding year, is to ensure the user will always see a reasonable output (0% - 100%), no matter what names the input is, so they will not be disappointed with an error.

Choosing the lowest possible rank is because that is the best estimate of the rank, given that names that are not ranked would be rare and low on occurrence if not none.