

Cycle 2 Requirements

Numbers in bold indicate top priority for this cycle.

Functional Requirements:

1. The system shall have searching functionality to allow the user to search for existing persons in the database. If the person being searched isn't in the database, the user can then add them to the database.
2. The system's search functionality shall allow the user to search up persons by first name, last name, or both, at a bare minimum. Additional methods, such as searching by partnering relationship(s), familiar relationships, etc. can be considered as development continues.
3. The system shall allow the user to add new persons to the database by first and last name. If nothing other than their first and last name is known, the user will still be able to add them.
4. The system shall allow the user to create new relationships (familial and/or partnering) between existing persons, if any new ones are discovered.
5. The system shall include functionality to find their sibling(s), parent(s), grandparent(s), children, and grandchildren of any given person. For example, the user can search up a person, and the system will show them the relevant data asked by the user.
6. The system shall allow the user to set a start and end date for a partnering relationship, whether it be during the addition of a new person, or for existing persons within the database.
7. The system shall allow the user to update existing relationships (familial and/or partnering) between existing persons, once more details are discovered (such as additional partners, who their father was, etc.). **(NOT IMPLEMENTED)**
8. The system shall allow the user to record any new children either during the creation of a new partnering relationship or updating an existing one. **(PARTIALLY IMPLEMENTED)**
9. The system shall allow the user to record any new children either during the creation of a new familial relationship or updating an existing one. **(PARTIALLY IMPLEMENTED)**
10. The system shall be able to determine if two people are related through recognition of a common ancestor. **(NOT IMPLEMENTED)**

Non-functional requirements:

1. The system shall be built using Java 11 (at a minimum).
2. The system shall be developed using the IntelliJ IDE.
3. The system shall be connected to the team's GitHub repository.
4. The system shall be broken up into different modules (such as classes) to allow team members to work on different parts simultaneously. These separate, working parts will all piece together in the end to create the final product.
5. Classes shall be broken up into different methods, and these different methods will be worked on by various members of the team.

6. Any work done by each member shall be done on their own branch, and upon pushing their changes, will create a pull request, where the rest of the team can inspect and audit their code.