

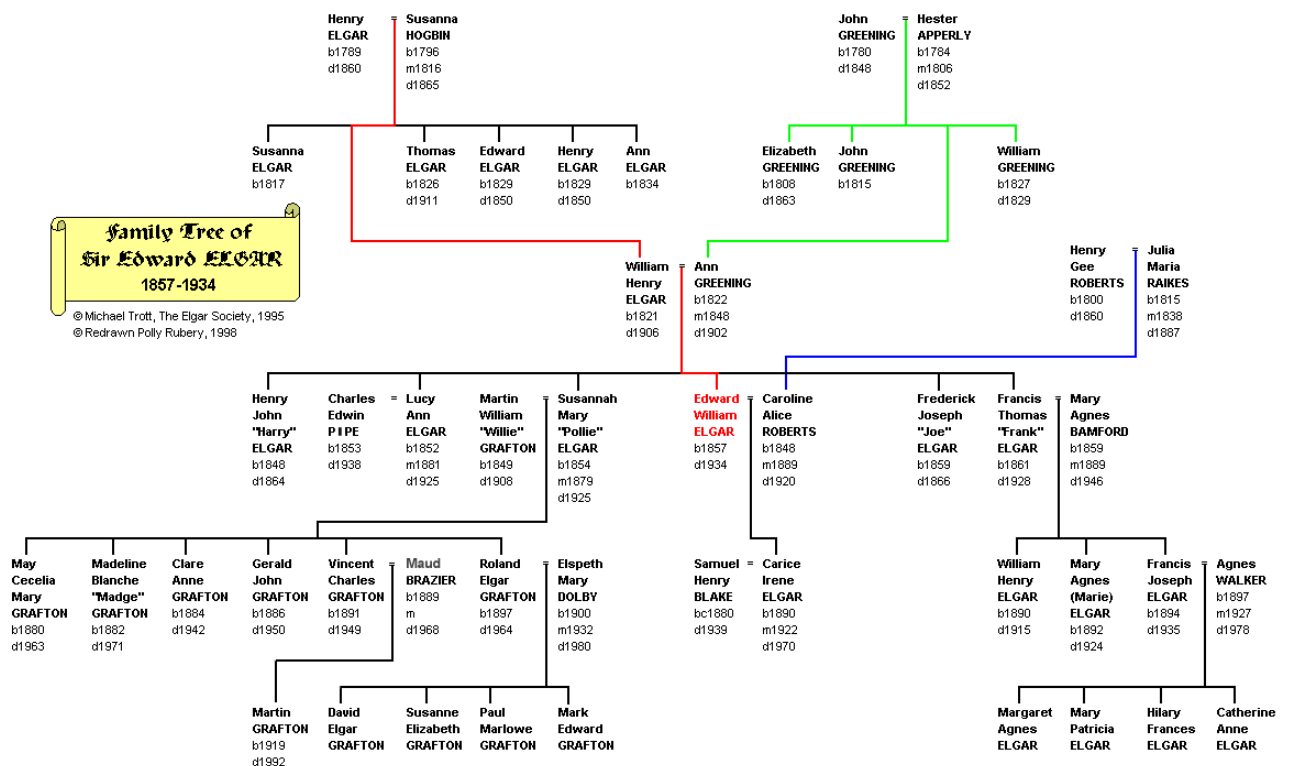
Genealogy App Functional Needs Statement

User / Customer Input:

Implement an app that keeps track of relationships between people and family genealogies or family trees. Family trees allow the user to determine ancestors and descendants of any person. Often family trees are incomplete especially during research to create them (there may be many people in them, some connected to each other and others not yet or ever connected). This app should allow storage and searching of arbitrarily complex sets of people partially connected into one or more family trees.

Since children result from partnering between the child's biological parents, the app includes records of partnership and any offspring associated with that partnering. Children are related to both their biological parents and parents may be in more than one partnership at the same or different times. It is required to trace all children of each person across all partnerships. It is required to trace upwards from any child to all their biological ancestors. It is required to trace downwards from any person to their descendants.

This picture may help you to visualize the kind of data the app needs to work with; it shows research into the families that joined together in the middle of the chart (William and Ann).



Here some things the app should do (possibly “Functional Requirements”):

1. Determine if a given person is known to the app. Provide at least look up by name (names do not need to be unique as they often repeat in families across generations). Likely useful to be able to search by family name, given names, date ranges, etc.
2. Add a person to the app. It may be necessary to connect them to their parents or children if these are already known in the app (update or add partnering).
3. Read in a text file of current people, their family relationships, and their partnering history. Upon request save any changes made back to a text file. We'll provide you an example text file to work with and the format.
4. Record the start and end of a partnering relationships.
5. Record a new child(ren) of an existing or new partnering.
6. For any person, find their parents, grandparents, etc. (for all those known by the app).
7. For any person, find their children, grandchildren, etc.
8. For any person, find their first cousins (first cousins have the same grandparents).
9. For any person find their current partner(s).
10. For two people determine if they are related. Two people are related if they have a common ancestor somewhere “up” their family trees.

We picked you as the implementation team since you are ACE software engineers. Here's some “Non Functional” requirements we know are important (and we trust you to add more):

1. Use Java v11 and IntelliJ for your program development. Save your work in the course GIT repository for your team.
2. Structure the system so that the user interface is separate from the logic and searching functions. Ideally the user interface can be implemented several different ways (text I/O, graphics, mobile device, big computer, smart watch, smart classes, voice access....; plan for this but only do what you and your team wish for the actual GUI).

3. Have a well-structured functional decomposition of the app into separate parts. This decomposition should support separate development of key components by individual programmers.
4. We will pay your team and decide to continue to hire you as developers based on seeing quick progress on versions of the app; so, you can decide where to start what to do first, what to add when. You pick what features to do, how to be able to quickly do something relevant and what order to add things. Each version should take 3 or 4 weeks to create.

Simplifications, Suggestions, and Assumptions:

1. A name is required for every person in the app; however, they need not be unique. Think about implementing sophisticated name searching (e.g. middle name, just middle initial, no middle name given, common misspellings, etc.) If appropriate use another technique to provide a unique key for each person (e.g. SSN, ID number, Public Key Signature, etc.)
2. Consider some way to identify a partnership between people in a unique way (e.g. registration number, place+date+time stamp, etc.)
3. Every child has exactly two biological parents; do not need to track other people associated with a family or children. However, zero, one, or two of the biological parents may be known to the system at any time. A child with no known parents is, of course, just another person in the app, possibly awaiting getting connected to parents, descendants, etc.
4. Partnerships between people may have formal start and end dates (as marriages do for legal purposes). Have an option to record these when known.
5. All people in the app may be either or both children and parents. Perhaps it's best to just think about people and partnerships.
6. Other ideas, suggestions, changes? Just let us know you want to discuss.