Family Tree

**XTeam No:** x190

**Team Members:** Nihir Singh : nsingh44@wisc.edu, Allison Seigler : aseigler@wisc.edu, Ryan Demar : rdemar@wisc.edu, YC Qiu : yqiu56@wisc.edu

Problem:

Often times, someone forgets a birthday or another important event occurring once a year of a

family member and this may lead to tensions in family relations, guiltiness, and various

other negative outcomes. The proposed program would make sure you remember the important date, and

whose special day it is!

Primary StakeHolder:

Anyone can use this application for their benefit, as it can remind anyone about the important dates. The user will benefit from the application as he/she will get reminded about the various important events and the name of the person and

how the person is related to the user.

Graphical User Interface:

Example of screen when the program is opened for the first time:

A close up of text on a whiteboard

Description automatically generated

After name is entered, the user is prompted to enter a CSV file as shown:

A close up of text on a whiteboard

Description automatically generated

If there is an error loading the CSV file, this screen will appear:

A close up of text on a white background

Description automatically generated

After data is successfully imported, the screen will look like this:

A close up of text on a whiteboard

Description automatically generated

The options on the left are buttons that will bring you to the corresponding pages above.

After the first time opening the app, this will be the default screen. The text in the main area would change automatically based on the day.

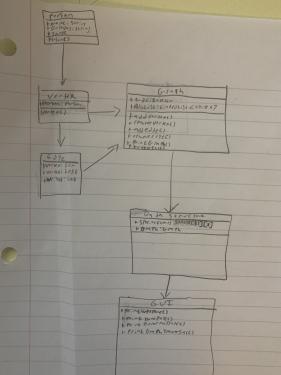
Data Structure:

We can use a graph (see more information in <https://docs.oracle.com/middleware/11119/jdev/api-reference-esdk/javax/ide/util/Graph.html>) to store different people as vertices and their relation to other people as edges. A Person object can be stored at each vertex, which would include a name, age, birthday, and possibly an image file. Graph traversals would always start at the user's vertex and find the vertex of the desired person. Once that vertex is found, the shortest possible path would be computed and printed out. As each edge from the user to the person is traversed, that edge's label (person's relation to other person) would be printed. An array of arrays can be used to store special events at each day. When the program is opened, the people associated with the current day would be searched for in the tree to begin the process.

eg: Today is John's (user), Dad's (Peter) , Sister's (Susan) , Husband's (Griffin), Brother's (Donald) Birthday !!!

Input Data File Format: A CSV file could be used, which specifies the family members, their relationship to other family members, and their birthdays.

Class Map:



This is our potential class map for the program with a person, vertex, edge, graph, data structure, and GUI class all labeled with their respective fields and methods

Output Example

(see GUI images for more):

start OUTPUT

Good Morning User,

Today is Donald's Birthday (March 26)

How are you related ?

Donald, brother of Griffin, husband of Susan, sister of Peter, Father of User.

Hope you have a nice and an amazing day :)

end OUTPUT

Milestones:

1. Implementing a graph to store the information, with edges that can store a relation and vertices that can store a person. Traversals must be able to return the shortest path and print out relations between people in the correct order. Additions and deletions must be successful. – before Apr 18

2. Implementing Date (see more information in <https://docs.oracle.com/javase/8/docs/api/java/util/Date.html>) and Time functionality so that birthdays corresponding to the current day are automatically displayed. Array storing special days associated with the current day must be implemented correctly and data must be pulled from the graph correctly. – before Apr 30

3. Making sure the GUI works as per the agreed drawings. – before May 8

Tasks:

1. Writing graph implementation, including insertion and deletion of vertices, insertion and deletion of edges, and basic search.- Allison

2. Writing method to correctly print out order of relationships between people. - Allison

3. Writing array of days and events stored in each day, making sure it can be used to print people's birthdays from the graph correctly based on a day. – Nihir

4. Using Date and Time functionality and the array of arrays to automatically print events corresponding to the current day – YC

5. Creating the GUI and making sure it works as described in the drawings.- Ryan

6. Making a sample CSV input file for the application. - YC