Phase 1, Part 2- Eunbie Coe, Francesca Gazzolo, Jessie Xiao

2. Technical Report

2a. List of ADTs

- **ADTs: Graph, Hashtable**
  - We would implement the Graph ADT to create an instance of a family tree. This graph would store information on a family with people and events that happened in history. This ADT is justified because it both visually and logically aligns with our project. We are using a graph instead of a Tree ADT because there are two roots (two parents) instead of just one root as in a tree.
  - A Hashtable would be implemented in the Family class and the FamilyTree class. In the Family class, the Hashtable is used to store the family member's name as the key and the details of the family member as a Person object as the value. In the FamilyTree class, the family name is the key and the family instance is the value. This ADT is justified because it helps store and call on specific details of a key.

2b. List of Classes

- **Classes:** Person, Family, Event, SingleEvent, Family Tree, Family Tree GUI
  - Person: Holds an instance of a person in a family.
  - Family: Holds a collection of Person objects to establish relationships and create a family. This is where the graph ADT and Hashtable ADT will be.
  - SingleEvent: Holds an instance of a notable event in history. Will include the event's name, time period, and description.
  - Event: A vector that contains all events the family was involved with with SingleEvent type.
  - Family Tree: Instance of a family tree. A Hashtable ADT will also be used here.
  - Family Tree GUI: Where all the GUI panels are created.

2c. List of methods

- **Person**
    - Class Variables
        - Birthdate: String
        - Name: String
        - Family Name: String
        - Time period: String
        - Ethnicity: String
        - Gender: String
        - Way they died: String
        - Reign: String
        - Position: String
        - Events involved in: Vector of StringEvent events
    - Methods:
        - Getters and Setters for class variables
        - toString() method
        - readPerson(): Read the person txt file
            - readEvents(): Helper method to read the list of events the person was involved in
- **Family**
    - Class Variables:
        - Number of people: String (String because of of reading txt file)
        - Time period: String
        - Collection of all persons in family: Hashtable
        - Bio: String
        - Graph: Adjacent Lists Graph of Person objects
    - Constructor:
        - Initializes graph and other instance variables
        - Reads from person txt file
        - Reads from bio txt file
    - Methods:
        - Getters for class variables
        - contains(): check if contains vertex
        - addParent(): adds an arc
        - addSpouse(): adds an edge
        - getDescendants(): gets successors in the Family Tree
        - getAncestors(): gets predecessors in the Family Tree
        - getSpouses(): returns spouses
        - getGraph(): returns AdjacentListGraph

        - isSpouse(): checks if edge
        - addArcs(): add all arcs between parents, children, and spouses
        - getSource(): returns person who is the source
- **SingleEvent**
    - Class Variables:
        - Name of event: String
        - Time period of event: String
        - Description of event: String
    - Methods:
        - Getter and setter methods
        - toString()
- **Event**
    - Class Variables:
        - Name of event: String
        - Time period: String
        - Brief description: String
        - Collection of all events the family was involved in: Vector of SingleEvent type
    - Constructor:
        - Reads from event txt file
        - Initializes the vector and other instance variables
    - Methods:
        - getEvents(): returns the collection of all events
        - getEvent(): returns single event
        - contains(): A method that tests if the collection of all events contains an event that is an argument.
- **FamilyTreeGUI**
    - Main Method
        - Sets up main JFrame for the GUI
        - Adds tabbed panels Family, Biography, and Events to the main JFrame
        - Combines all tabbed panes together and sets them to be visible
- **FamilyTree**
    - Class Variables:
        - Families: Hashtable
        - Current Family: Family
        - Tudor Events: Event
    - Constructor
        - Initializes the Current Family as the Tudors
        - Puts the Tudors in the Families Hashtable

        - Adds arcts to Tudors
        - Initializes the events in Tudor Events
    - Methods:
        - getCurrentFamily(): returns the current family we are examining
        - getFamily(String familyName): returns and object of type Family searched by family name
        - size(): returns the number of families stored in the hashtable
        - getEventList(): returns the list of events, Tudor Events