Testing Plan

Using JUnit, I will develop test cases for our app. The classes to be tested include:

1. GeneDataBase, with class GeneDataBaseTest
    a. A GeneDataBase object stores the database and primarily deals with relationships between Person objects in the database.
2. Person, with class PersonTest
    a. A Person object stores specific information about an individual person.

GeneDataBaseTest Cases

1. Reading people in to the database from the file
    a. public void testPlantedPeople()
2. Creating marriage/partnership information:
    a. Based on initial data read in from input file
        i. public void testPlantedPartnerships()
    b. Based on new information (simulating the GUI)
        i. public void testCreatePartnership1()
3. Finding the parents of a given person in the database, including parents resulting from new marriages
    a. Based on data read in from input file
        i. public void testPlantedParents()
4. Finding the children of a given person in the database, including children resulting from new marriages
    a. Based on data read in from input file
        i. public void testPlantedChildren()

5. Finding the siblings of a given person in the database, including siblings resulting from new marriages
    a. Based on data read in from input file
        i. public void testPlantedSiblings()
6. Finding the grandparents of a given person in the database, including grandparents resulting from new marriages
    a. Based on data read in from input file
        i. public void testPlantedGrandparents()
7. Finding all people in the database with a particular first name and surname
    a. public void testFindExactNameID1()
8. Finding all people in the database with a particular surname
    a. public void testFindExactSurnameID1()
9. Finding all people in the database with a particular first name
    a. public void testFindExactFirstNameID1()

PersonTest Cases

1. Verifying that a living person's age is correct
    a. Who has a date of birth
        i. public void testDateAge1()
    b. Who does not have a date of birth
        i. public void testDateAge4()
2. Verifying that a dead person's age is correct
    a. Who has a date of birth
        i. public void testDateAge3()

    b. Who does not have a date of birth
        i. public void testDateAge2()
3. Verifying that a person's name is correctly registered
    a. public void testName1()
4. Verifying that a person's ID is correctly registered
    a. For person with a given ID
        i. public void testID1()
    b. For person with a blank ID
        i. public void testID2()
5. Verifying that a person's birthplace and death place are correctly registered
    a. public void testBirthDeathPlace1()
6. Verifying that a person's marriage details are correctly registered
    a. People who has not yet been married
        i. public void testMarriageDetails1()
    b. Couple where one partner was previously married before
        i. public void testMarriageDetails2()
7. Verifying that a person's parents' relationship ID is correctly registered
    a. For someone with a parent relationship ID
        i. public void testParents1()
    b. For someone without a parent relationship ID
        i. public void testParents2()
8. Verifying that a person's gender is correctly registered
    a. For someone who is female

    i. public void testGender2()

b. For someone who is male

    i. public void testGender1()

c. For someone who does not have a gender

    i. public void testGender3()