Rushi Samirbhai Patel RS397441@dal.ca
B00886157

Final Project

Genealogy Media Integration System

Final Document

Overview

This document consists of all the changes that were made from Milestone 4 (mainly database changes and some assumption changes). I have also created JUnit test plans as part of white box testing plan which will be submitted along with all the java files. There is a main method for a better reference and for flexibility to run the code.

Testing Plan

Along with all the tests in my previous milestones, I am submitting my JUnit test plans which includes all the business logic tests along with all the class tests.

Database Changes

I have added some tables, added 'id' columns from children,dissolution and partnering tables, removed tags and individualInMedia columns from mediaArchive table, created separate tables for tags and media people (mediatag and mediapeople) respectively, and added primary as well as foreign keys in my database. I will be attaching the sql file along with this document which would also contain some sample data for your reference. I will also provide only the create table statements for your ready reference at the end of this document itself too for your ready reference.

Assumptions

I have made some assumptions for this project which include but are not limited to:

1. For a person, multiple occupations are not allowed.
2. Partnering and Dissolution relations can be recorded even after their death.
3. Year entered in the system must follow the format 'YYYY-MM-DD'.
4. While externally providing any person or media file object for testing, its id must be set at all times.
5. Location and city are considered as the same however, users can provide both location and city in location parameter and findMediaByLocation with either of the two or both.

6. A child can be recorded even if the person is not married (adoption) but will still notify the user that the person does not have a partner.

Code Design And Algorithm Changes

From my previous implementation plan, I have made some changes to implement findRelation, ancestors and descendants functionalities. To find the relation, I have maintained my database and have fetched information from only the children table where I am storing id's of both parent and child. These id's are stored in a map having key as parent id and value as the parent's children (id's). With the help of two arayLists and populating them and verifying if the same id is present in both the list, I am determining the id of the nearest common ancestor as well as its cousinship and degree of removal. Likewise for ancestors and descendants, I am using the same table from the database - children table. With the help of both the id's, I am calculating generations based on the number of children of the person and iterating the loop till the generations match the level of generation provided in the method parameter. I am doing so with the help of three array lists (clearing the third at the end of each generation and re-populating the second to find their children which would eventually store all the descendants in the first list. With the same algorithm, I am finding the ancestors.

Files and External Data

I have added some files in the project from the ones I had reported in my previous milestone. They are:

1. **BiologicalRelation.java** - This class is used to denote the relation of two people.
2. **FileIdentifierUtilTestPlan.java** - Test Plan for FileIdentity
3. **Geneology.java** - This class is to assist the geneologist by fetching data from the database based on their requirement.
4. **GeneologyException.java -** This class is used as a custom exception to handle bad inputs in the program.
5. **GeneologyTestPlan.java -** Class to test Geneology functionalities.
6. **IGeneology.java -** Interface for Geneology class.
7. **PersonIdentityTestPlan.java -** Test Plan for PersonIdentity.

Rushi Samirbhai Patel RS397441@dal.ca
B00886157

Updated SQL Queries to replicate my schema and database:

I have attached sql files along with this document as well for your reference. I have also attached exported data from my sql database with some data for your reference.

/*Creating Schema for the project*/

```
CREATE DATABASE `finalproject` /*!40100 DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_0900_ai_ci */ /*!80016 DEFAULT ENCRYPTION='N' */;
use finalproject;
```

/*Creating family tree database*/

```
create table FamilyTree (id int primary key,individualName varchar(90) not null,dateOfBirth date,locationOfBirth varchar(90), dateOfDeath date, locationOfDeath varchar(90), gender varchar(90), occupation varchar(90), referencesToSourceMaterial varchar(90), notes varchar(90), partner varchar(90), dissolution varchar(90));
```

/*Creating family tree database*/

```
create table MediaArchive (id int primary key not null, fileName varchar(90) not null, dateOfMedia date, location varchar(90));
```

/*Creating table to map references of individual*/

```
create table ReferencesOfIndividual (id int , individualName varchar(90), referencesToSourceMaterial varchar(90), foreign key(id) references familytree(id));
```

/*Creating table to map notes of individual*/

```
create table NotesOfIndividual (id int, individualName varchar(90), notes varchar(90), foreign key(id) references familytree(id));
```

/*Creating table to record child*/

```
create table Children (parent int, child int,foreign key(parent) references familytree(id),foreign key(child) references familytree(id));
```

/*Creating table to record partnering relations*/

```
create table Partnering (partner1 int, partner2 int,foreign key(partner1) references familytree(id),foreign key(partner2) references familytree(id));
```

/*Creating table to record dissolution relations*/

```
create table Dissolutions (partner1 int, partner2 int,foreign key(partner1) references familytree(id),foreign key(partner2) references familytree(id));
```

Rushi Samirbhai Patel RS397441@dal.ca
B00886157

/*Creating table mediatag*/
create table mediatag (id int,fileName varchar(90), tag varchar(90), dateOfMedia date,foreign key(id) references MediaArchive(id));

/*media people updated created table*/
create table mediapeople (id int,fileName varchar(90), individualName varchar(90), dateOfMedia date,foreign key(id) references MediaArchive(id),foreign key(id) references familytree(id));

Please contact me in case of any concerns.

Thank You!