Prachi Prakashbhai Raval
B00883324
pr647062@dal.ca

12th November ,2021

# CSCI 3901 - FINAL PROJECT

MILETONE 2:
BLACKBOX TESTS AND PLAN OF FEATURE DEVELOPMENT

**Index**

Feature Development Plan:

1. Family Tree Management - Record family information and store in tree and database simultaneously:

   11th Nov – 15th Nov

   - Add person to family tree person name
   - Record person attributes like DOB, birthplace, gender, occupation
   - Record Notes and references

   16th Nov – 20th Nov

   - Record Relationships like parent child, partnering and dissolution
   - Build Tree using Biological Relationship Information

2. Media archive management – Media Files information and Individuals present in media

   20th Nov- 26rd Nov

   - Store photos and video location
   - Store media files attributes like date, location, city, year, etc.
   - Store tags in media.

   26th Nov-31st Nov

   - Store individuals present in a media file.

3. Reporting – Export your information to answer genealogist's questions also store in some data structures to return them.

   31st Nov- 5th Dec

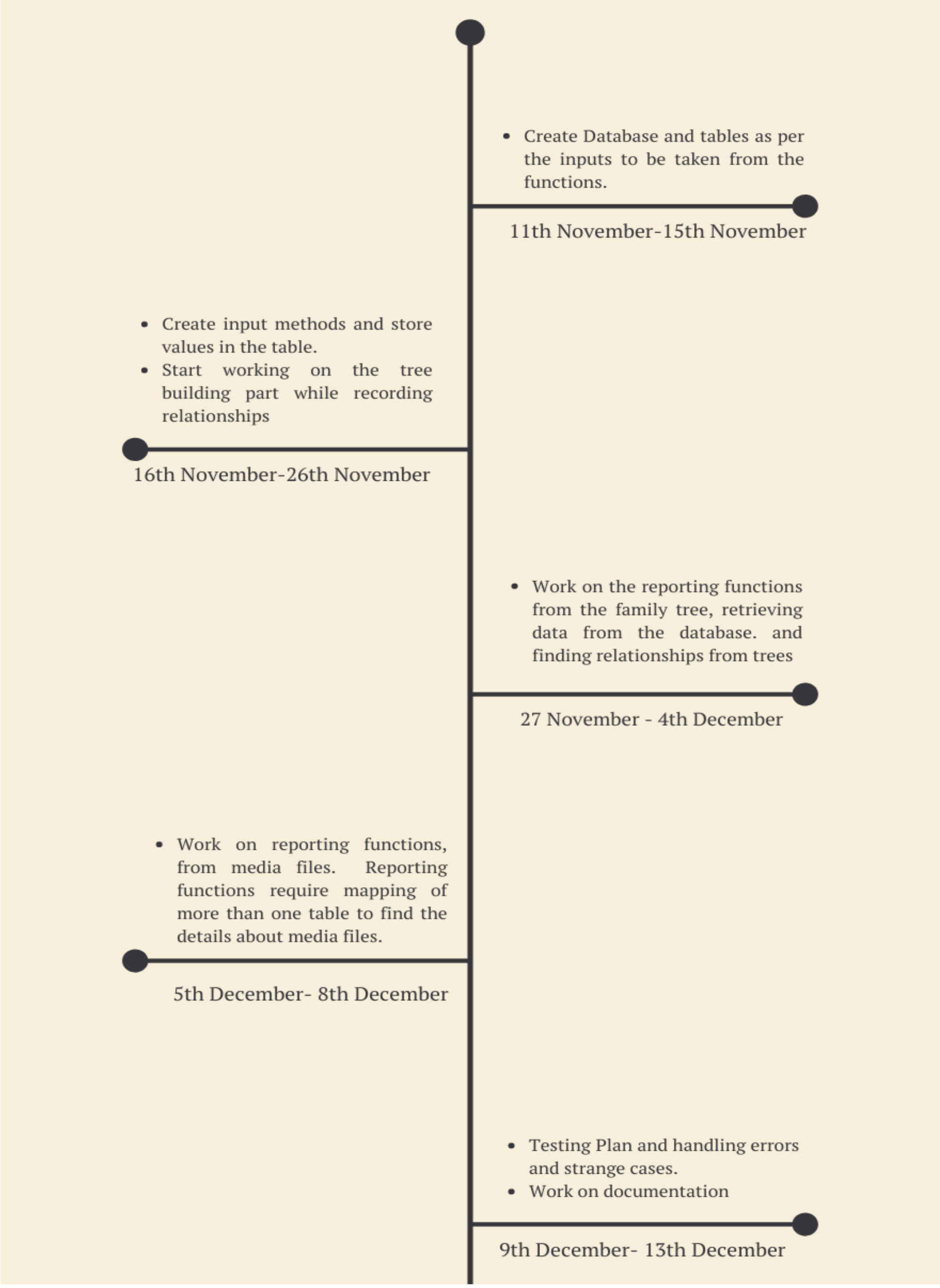
   - Export your information to answer genealogists' questions.
   - Find person and media file.
   - Find List of notes and references.
   - Find biological relationship between 2 persons.
   - Find set of descendants and ascendants.

   5th Dec- 9th Dec

   - Find set of media files with given tags between particular start and end date.
   - Find set of media files with given location between particular start and end date.
   - Find set of media files that include any of individuals given in the list of people whose dates fall within the date range.
   - Find set of media files that include the specified person's immediate children.

# Timeline chart

## (Approximate range of dates to work on paricular feature)



- Create Database and tables as per the inputs to be taken from the functions.

11th November-15th November

- Create input methods and store values in the table.
- Start working on the tree building part while recording relationships

16th November-26th November

- Work on the reporting functions from the family tree, retrieving data from the database. and finding relationships from trees

27 November - 4th December

- Work on reporting functions, from media files. Reporting functions require mapping of more than one table to find the details about media files.

5th December- 8th December

- Testing Plan and handling errors and strange cases.
- Work on documentation

9th December- 13th December

## Black Box testing:

- Equivalence Partitioning - It is a software testing technique or black-box testing that divides input domain into classes of data, and with the help of these classes of data, test cases can be derived.
- Boundary Value Analysis - Boundary Value Analysis (BVA) is a Black-Box testing technique used to check the errors at the boundaries of an input domain.
- Decision Table Testing - This is a systematic approach where the different input combinations and their corresponding system behavior (Output) are captured in a tabular form.
- Error guessing- Error guessing is a method of black box testing that relies solely on the previous experience of the tester. The software tester uses their past experience to determine where errors in the software may be.

## Equivalence Partitioning:

- Person id is the primary key for the person information. It can be tested by dividing into two groups.
  PersonId>1 and PersonId >1

- Generations have a lower limit of 1. They can always be positive integers. Generations have an upper limit of the size of the family tree.
  Generations>= Max Level of Tree.
  Partitioning the test cases.
  Max tree level>Generations>=1 and Generations<1

- Attributes in family tree and media archive:

  For Occupation, Name, Location, Occupation, Media Tags

| Valid Equivalent cases | Invalid Equivalent cases |
|---|---|
| 1: Contains numbers 2: Contains lower case letter<br>3: Contains upper case letter<br>4: String length between 2 - 45 | 1: non-ASCII character<br>2: String length>45<br>3: String length<1<br>4: Contains special character |

For Date of birth, date of death in family tree and for date and year in media archive. (Taken as integer for simplicity)

| Valid Equivalent cases | Invalid Equivalent cases |
|---|---|
| 1: Contains numbers | 1: Contains alphabets<br>2: Contains special character |

- String such as notes and references can be stored in varchar(max) which has a very high limit of characters storage. Thus, it can store paragraphs and sentences as maximum length of the string.
  Test portioning
  1<=String<2 147 483 647 characters and String< =2 147 483 647 characters

| Valid Equivalent cases | Invalid Equivalent cases |
|---|---|
| 1:Contains numbers<br>2:Contains lower case letter<br>3:Contains upper case letter<br>4:Contains special character<br>5:Sting length between 2 - 2 147 483 647 | 1:Non-ASCII character<br>2:String length>2 147 483 647<br>3: String length<1 |

- While noting the parent child relationship and building the tree. Testing can be divided into:

  Parent has one child.

  Parent has more than one child.

- Individuals in media file, testing can be partitioned into two groups
  File contains 1 person. (Size of list =1)
  File contains more than 1 person (Size of list >1)

## Boundary Value Analysis:

- addPerson
  - 1 character student's name
  - Must be within the length if defined
- recordAttributes for family Tree
  - Examples of attributes are "date of birth", "gender", and "occupation"

  - Date of Birth- Date must between 1 and 31 and month between 1 and 12
  - Gender- must be Male or Female
  - Occupation -1 character occupation
- recordAttributes for media Archive
  - Examples of attributes are “year”, “date”, and “city”
  - Date must be between 1 and 31 and month must be between 1 and 12
  - City 1 character city name

## Decision Table Testing:

| **Methods** | **Conditions** | **Return Value** |
|---|---|---|
| Record Attributes, notes, references, tags, media location | String length is appropriate and not null or empty | True |
| Record Attributes, notes, references, tags, media location | String null, empty or not stored in database | False |
| Record child | Parent child relation is stored in database. | True |
| Record child | Parent or child does not exist in family tree. | False |
| Record Partner | Person1 and Person2 has no symmetric relationship existing, then their partnering stored to database. | True |
| Record Partner | Person1 or Person2 already has one symmetric relationship existing. | False |

| | | |
|---|---|---|
| Record Dissolution | Person1 has partner with person 2 | True |
| Record Dissolution | Person1 does not have partnering with person 2 | False |

## Error Guessing:

- Handling null values in String / text fields.
- Handling negative values for integers.
- Handling Empty String.
- Handling inputs of string greater than the datatype in database.
  All these conditions should return error and should be handled by exception handling or if/else conditions.

**References:**

- http://www.airccse.org/journal/ijsea/papers/1011ijsea04.pdf
- https://www.softwaretestinghelp.com/black-box-testing/