## Black Box Tests

1. Adding person
- Person name is an empty string
- Person name is made up of 1 character
- Person name is made up of maximum allowed characters
- Person name is between 2 and maximum allowed characters
- Person name is made up of more than maximum allowed characters
- Person with the same name added for the second time

2. Recording person attributes
- Record when the person does not exist
- Date of birth format is incorrect
- Date of birth is a future date
- Gender is not in the correct format
- 1 character occupation name
- Maximum allowed characters occupation name
- Occupation name is an empty string
- Occupation name is between 1 and maximum allowed characters

3. Recording reference
- Reference for that individual does not exist
- 1 reference exists for that individual
- Many references exist for that individual
- Record when the person does not exist
- Reference is an empty string
- 1 character reference string
- Maximum allowed characters reference string

4. Recording note
- Note for that individual does not exist
- 1 note exists for that individual
- Many notes exist for that individual
- Record when the person does not exist
- Note is an empty string
- 1 character note string
- Maximum allowed characters note string

5. Recording child
- Parent does not exist
- Child does not exist

- Parent is partnered
- Parent is single
- After dissolution
- Child already linked to different a parent(parents)
- Child already linked to the same parent
- Child already linked to the partner
- Child is partnered
- Child is partnered and added to a parent linked(ancestor) to the partner

6. Recording partnering
- Partner 1 does not exist
- Partner 2 does not exist
- Partner 1 already has a partner
- Partner 2 already has a partner
- Partner 1 partnering with partner 2 who had dissolution
- Partner 2 partnering with partner 1 who had dissolution
- Partnering between individuals having common ancestor
- Partnering between individuals having cousin relationship
- Partnering between individuals having ancestor/descendant relationship.

7. Recording dissolution
- Partner 1 does not exist
- Partner 2 does not exist
- Partner 1 and Partner 2 are not partnered
- Partner 1 and Partner 2 are partnered
- Dissolution twice between two individuals

8. Adding media file
- File location is an empty string
- File location is made up of 1 character
- File location is made up of maximum allowed characters
- File location is between 2 and maximum allowed characters
- File location is made up of more than maximum allowed characters
- Media with the same location added for the second time

9. Recording media attributes
- Record when the media does not exist
- Date format is incorrect
- Date is a future date
- 1 character city name
- Maximum allowed characters city name

- City name is an empty string
- City name is between 1 and maximum allowed characters

10. Recording people linked to a media
- Media does not exist
- No people passed
- One person passed
- Many people passed
- Out of many people, one person passed does not exist

11. Tagging media
- Tag does not exist
- Tag already exists
- Media does not exist
- Tag exceeding maximum allowed character limit
- Tag is already linked with a different media
- Tag is not linked with any other medias
- Tag is an empty string
- 1 character tag
- Many characters tag

12. Finding person
- Person does not exist
- Person exists

13. Finding media file
- Media does not exist
- Media exists

14. Finding an individual's name
- Person exists
- Person does not exist

15. Finding media file
- Media exists
- Media does not exist

16. Finding relation
- Person 1 does not exist
- Person 2 does not exist
- Both person 1 and person 2 exist
- Person 1 and Person 2 are partners

- Person 1 and Person 2 are related
- Person 1 and Person 2 are not related

17. Finding descendants
- Generations is 0
- Generations is a negative value
- Generations is more than the actual available generations
- Generations is 1
- Person does not exist
- Person exists
- Person has a single parent
- Person has both the parents
- After dissolution between parents
- After dissolution between parent and then one of the parents partnering with a different person

18. Finding ancestors
- Generations is 0
- Generations is a negative value
- Generations is more than the actual available generations
- Generations is 1
- Person does not exist
- Person exists
- Person has a single parent
- Person has both the parents
- After dissolution between parents
- After dissolution between parent and then one of the parents partnering with a different person

19. Finding notes and references
- Person does not exist
- Person exists
- Person has no notes associated
- Person has a single note associated
- Person has many notes associated
- Person has no references associated
- Person has a single reference associated
- Person has many references associated

20. Finding media by tag
- Tag exists
- Tag does not exist
- Tag is an empty string
- Start date is invalid

- End date is invalid
- Valid dates
- Start date is an empty string
- End date is an empty string
- Tag not associated with any media
- Tag associated with many medias

21. Find media by location
- Location exists
- Location does not exist
- Location not associated with any media
- Location associated with many medias
- Location is an empty string
- Start date is invalid
- End date is invalid
- Valid dates
- Start date is an empty string
- End date is an empty string

22. Finding individuals media
- Start date is invalid
- End date is invalid
- Valid dates
- Start date is an empty string
- End date is an empty string
- One of the persons passed does not exist
- One person passed
- Many persons passed
- Not a single person passed

23. Finding biological media
- Person does not exist
- Person is partnered
- Person is partnered after dissolution
- Person has a child
- Person has many children
- Person has no children
- Children have no media associated
- Children have media associated

## Feature Development Plan

1. Create conceptual, logical and physical models for storing the family tree information and media entities.

| Tasks | Entities | Tools | Planned date of completion |
|---|---|---|---|
| Design conceptual/logical model | person, person_attribute, note, reference, partner, parent_child, media, media_attribute, people_media, tag | Draw.io | 12/11/2021 |
| Design physical model: Define tables and column constraints | | MySQL Workbench | 14/11/2021 |
| Design physical model: Develop ERD, Normalize using iterative approach, Forward engineer the model | | | 17/11/2021 |
| Integrate Genealogy database with the java code | NA | Java, Eclipse, MySQL connector | 19/11/2021 |

2. Work on the java classes.

| Tasks | Planned date of completion |
|---|---|
| 1. Create interfaces called FamilyTree, MediaManagement, and Reporting.<br>2. Create a class called Geneaology implementing above interfaces.<br>3. Implement FileIdentifier class.<br>4. Develop SQL queries for adding media, recording media attributes, and tagging media.<br>5. Implement methods for storing media information in the database:<br>• addMediaFile<br>• recordMediaAttributes<br>• tagMedia | 21/11/2021 |
| 1. Implement PersonIdentity class.<br>2. Develop SQL queries for adding person, recording person attribute, recording reference and recording note.<br>3. Implement methods for storing an individual's information in the database:<br>• addPerson<br>• recordAttributes<br>• recordReference | 23/11/2021 |

| | |
|---|---|
| • recordNote<br>4. Implement the peopleInMedia method. | |
| 1. Implement data structure for storing family tree hierarchy<br>2. Develop SQL queries for recording child, recording partner, and recording dissolution.<br>3. Implements methods for defining relationships in the database and in the memory:<br>• recordChild<br>• recordPartnering<br>• recordDissolution<br>4. Implement methods to retrieve information from the database to build a family tree hierarchy. | 28/11/2021 |
| 1. Implement methods to find the most recent ancestor in the family tree<br>2. Develop SQL queries for finding a person, finding a media file, finding name, and finding a media file.<br>3. Implement reporting functions:<br>• findPerson<br>• findMediaFile<br>• findName<br>• findMediaFile<br>• findMediaByTag<br>• findMediaByLocation<br>• notesAndReferences | 02/12/2021 |
| 1. Implement reporting functions<br>• findRelation<br>• descendents<br>• ancestors<br>• findIndividualsMedia<br>• findBiologicalFamilyMedia | 09/12/2021 |