CSCI 3901 FINAL PROJECT – TEST PLAN

Instructor: Dr. Michael McAllister
Name: Nadipineni Hemanth Kumar
Banner ID: B00899473
Date: 2021-12-07

Test Plan:

Objective:

The fundamental goal of testing is to ensure that the system meets all criteria, including quality requirements (both functional and non-functional) and fit metrics for each quality requirement, as well as to satisfy use case scenarios and maintain product quality. Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented.

As a result, the technique employed is Analytical, in line with a requirements-based strategy, in which an analysis of the requirements specification serves as the foundation for planning, estimating, and developing tests. During exploration, test cases were built (Milestone-2 & Milestone-3). The sorts of tests are listed below.

- JUnit 5 was used for testing the application.

Testing Types:

1. Blackbox Testing:

   Black-box testing is a type of software testing that looks at an application's functioning without seeing inside its internal structures or workings. This test approach can be used at all levels of software testing, including unit, integration, system, and acceptance. Specification-based testing is another name for it [1].

2. Whitebox Testing:

   White box testing is a type of testing that looks at the structure of a program and generates test data from the logic/code. Clear box testing, open box testing, logic-driven testing, path-driven testing, and structural testing are some of the alternative names for glass box testing [2].

3. JUnit Testing:

   A JUnit test is a method in a class that is solely for testing purposes. This is referred to as a Test class. Annotate a method with the @Test annotation to make it a test method. This method runs the code that is being tested [3].
   (jUnit test file is submitted along with code)

Testing Assumptions:

- Database will be available for reliable testing.
- VPN access will be provided to connect to the dal DB
- No environmental downtime during the tests.

Testing Requirements:

- The database hosted on a local pc or on dal servers.
- IntelliJ
- Java 11
- Junit 4.13.2
- JDBC connector

Blackbox Tests:

- Add a person when there are no persons in the database
- Add a person when there are multiple persons in the database
- An Empty name is given to a person while adding

- Null is given to a person
- Add a person when there's a person with the same name exists in the database
- Name with special characters and numbers in between (Alphanumeric names) is passed
- Record attributes to a non-existing person in the database
- Record the date of birth for an existing person
- Record the date of birth with year of birth
- Record the date of birth without year of birth
- Record the date of birth with month of birth
- Record the date of birth without month of birth
- Record the date of birth with day of birth
- Record the date of birth without day of birth
- Record the date of birth that is in future
- Record duplicate date of birth for a person with already existing date of birth in the database
- Record location of birth for an existing person
- Record the date of birth with alphabets in between when the date of birth is stored as a string
- Record duplicate location of birth for a person with already existing location of birth in the database
- Record the date of death that is after date of birth for a person with date of birth in the database
- Record the date of death that is before date of birth for a person with date of birth in the database
- Record the date of death with year of birth
- Record the date of death without year of birth
- Record the date of death with month of birth
- Record the date of death without month of birth
- Record the date of death with day of birth
- Record the date of death without day of birth
- Record the date of death that is in future
- Record the date of death with alphabets in between when the date of death is stored as a string

- Record duplicate date of birth for a person with already existing date of birth in the database
- Record the location of death for an existing person
- Record duplicate location of death for a person with already existing location of death in the database
- Record gender of an existing person in the database
- Record duplicate gender for a person with already existing gender in the database
- Record occupation of an existing person in the database
- Record duplicate occupation for a person with already existing occupation in the database

- Record reference material for a non-existing person in the database
- Record the first reference material for an existing person in the database
- Record the second reference material for an existing person in the database
- Record a reference material for an existing person with multiple references already existing to that person in the database
- Record a duplicate reference material that already exists for the person
- Record a note for a non-existing person in the database
- Record the first note for an existing person in the database
- Record the second note for an existing person in the database
- Record a note for an existing person with multiple notes already existing to that person in the database
- Record a duplicate note that already exists for the person
- Record a child relationship for a non-existing person in the database
- Record a child relationship for an existing person in the database
- Record multiple child relations for an existing person in the database
- Record a parent relationship for a non-existing person in the database
- Record a parent relation for an existing person in the database
- Record a second parent relation for an existing person in the database
- Record multiple parent relations for an existing person in the database
- Record a duplicate parent-child relationship when there is already existing one

- Create a parent-child relationship between two existing persons in the database where parent's date of birth (if exists in the database) is before the child's date of birth (if exists in the database)
- Create a parent-child relationship between two existing persons in the database where parent's date of birth (if exists in the database) is after the child's date of birth (if exists in the database)
- Create a parent-child relationship between two existing persons in the database where parent's date of death (if exists in the database) is after the child's date of birth (if exists in the database)
- Create a parent-child relationship between two existing persons in the database where parent's date of death (if exists in the database) is before the child's date of birth (if exists in the database)
- Record partnering relation between two persons non-existing in the database
- Record partnering relation between one person existing in the database and the other person not existing in the database
- Record partnering relation between two persons existing in the database
- Record partnering relation between two partners where there is already a partnering relation exists between them
- Record partnering relation between two partners where one partner already has a partnering relation with some other partner
- Record partnering relation between two partners where both partners having partnering relations with some other partners
- Record partnering relation between two partners where one partner's date of birth (if exists in the database) is after the date of birth (if exists in the database) of the second partner
- Record partnering relation between two partners where one partner's date of death (if exists in the database) is after the date of birth (if exists in the database) of the second partner
- Record partnering relation between two partners where one partner's date of death (if exists in the database) is before the date of birth (if exists in the database) of the second partner
- Record a dissolution of partnering relation between two persons non-existing in the database

- Record a dissolution of partnering relation between one person existing in the database and the other person not existing in the database
- Record a dissolution of partnering relation between two persons existing in the database
- Record a duplicate dissolution for two partners when the same dissolution already recorded in the database
- Record a dissolution of a partnering relation between two partners with partnering relation between them
- Record a dissolution of a partnering relation between two persons without a partnering relation between them

- Add a media file to the media archive with correct file location/path
- Add a media file to the media archive with incorrect file location/path
- Add a media file to the media archive with format specified (Photo/Audio/Video/etc.)
- Add a media file to the media archive with format unspecified
- Add a media file to the media archive when there are no media files existing in the media archive
- Add a media file to the media archive when there are multiple media files existing in the media archive
- Add a duplicate media file to the media archive that already exists in the media archive
- Add a media file when file location is null
- Add a media file when file location is empty

- Record media attributes for a file not existing in the media archive
- Record year that is in the past for a file existing in the media archive
- Record year that is in future for a file existing in the media archive
- Record date that is in future for a file existing in the media archive
- Record date that is in past for a file existing in the media archive
- Record a city name for a file, existing in the database, where the file was taken

- Record any other attributes corresponding to the media file existing in the database

- Record a list of people, existing in the family database, appear in the given media file with valid file identifier
- Record a list of people, non-existing in the family database, appear in the given media file with valid file identifier
- Record a list of people, existing in the family database, appear in the given media file with invalid file identifier
- Record a duplicate list of people as appearing in the given media file when there's already the exact list of people stored for the media file
- Record a list of people as appearing in the given media file when there's another different list of people stored for the media file
- Record a list of people as appearing in the given media file where a person's date of birth (if exists in the database) is after the file's date (if exists in the media archive)
- Record a list of people as appearing in the given media file where many persons' date of birth (if exists in the database) is after the file's date (if exists in the media archive)
- Record a list of people as appearing in the given media file where all persons' date of birth (if exists in the database) is after the file's date (if exists in the media archive)
- Record a list of people as appearing in the given media file where a person's date of death (if exists in the database) is before the file's date (if exists in the media archive)
- Record a list of people as appearing in the given media file where many persons' date of death (if exists in the database) is before the file's date (if exists in the media archive)
- Record a list of people as appearing in the given media file where all persons' date of death (if exists in the database) is before the file's date (if exists in the media archive)

- Record a tag for a media file with a file identifier non-existing in the media archive
- Record the first tag for a media file with a valid file identifier
- Record a tag for a media file when there is one or more tags already there for the file
- Record a duplicate tag for a media file with a valid file identifier
- Record an empty tag for a media file existing in the media archive
- Record a null tag for a media file existing in the media archive

- An empty string is passed as name to find a person
- null is passed as name to find a person
- Find a person when there is no person recorded in the family tree
- Locate an individual in the family tree when there is only one person in the family tree
- Locate an individual in the family tree when there is more than one person in the family tree
- A name that is not in the database is passed to find a person

- An empty string is passed as name to find a media file
- null is passed as name to find a media file
- Find a media file when there is no media file recorded in the media archive
- Locate a media file in the family tree when there is only one media file in the media archive
- Locate a media archive in the family tree when there is more than one media file present in the media file
- A name that is not in the media archive is passed to find a media file

- Find the name of an individual using ID and there's no person recorded in the database
- Find the name of an individual using ID and there's only one person recorded in the database

- Find the name of an individual using ID and there is more than one person recorded in the database
- An ID that is not present in the database is used to find the name

- Find the name of the media file using field and there's no person recorded in the database
- Find the name of the media file using field and there's only one person recorded in the database
- Find the name of the media file using field and there is more than one person recorded in the database
- A field that is not present in the database is used to find the media file

- Find relation between two persons X and Y when there is only one person ever recorded in the database
- Find relation between two persons X and Y when there is more than one person recorded in the database
- Find relation between two persons X and Y when both exist in the database
- Find relation between two persons X and Y when only one of them exist in the database
- Find relation between two persons X and Y when none of them exist in the database
- Find descendents of a person X for Z 'generations' number away when the person is not existent in the database
- Find descendents of a person X for Z 'generations' number away when the person is existent in the database
- Find descendents of a person X for Z 'generations' number away when there are no descendents available at that number of 'generations'
- Find descsendents of a person X when there is only one person recorded in the database
- Find the descendents of a person X when there is more than one person recorded in the database
- Find the descendents of a person X when there is no person ever recorded in the database

- Find ancestors of a person X for Z ‘generations’ number away when the person is not existent in the database
- Find ancestors of a person X for Z ‘generations’ number away when the person is existent in the database
- Find ancestors of a person X for Z ‘generations’ number away when there are no ancestors available at that number of ‘generations’
- Find ancestors of a person X when there is only one person recorded in the database
- Find the ancestors of a person X when there is more than one person recorded in the database
- Find the ancestors of a person X when there is no person ever recorded in the database
- Find notes and references when the given person is not existent in the database
- Find notes and references when the given person is existent in the database
- Find notes and references of an existing person in the database when there is no notes and reference linked to the person
- Find notes and references of an existing person in the database when there is one note and reference linked to the person
- Find notes and references of an existing person in the database when there are many notes and references linked to the person
- Find notes and references of a given person when there’s only one person ever recorded in the database
- Find notes and references of a given person when there’s more than one person ever recorded in the database
- Find notes and references of a given person when there’s no person ever recorded in the database

- Find the set of media files linked to a given tag and date range when the tag exists in the media archive
- Find the set of media files linked to a given tag and date range when the tag doesn’t exist in the media archive

- Find the set of media files linked to a given tag and date range when there are no media files exist in the given date range
- Find the set of media files linked to a given tag and date range when there are media files exist in the given date range
- Find the set of media files linked to a given tag and date range when the given start date is before the given end date
- Find the set of media files linked to a given tag and date range when the given start date is after the given end date
- Find the set of media files linked to a given tag and date range when the given start date is null
- Find the set of media files linked to a given tag and date range when the given end date is null
- Find the set of media files linked to a given tag and date range when the given start date and the given end date are null
- Find the set of media files linked to a given tag and date range when there's only one media file exists in the media archive
- Find the set of media files linked to a given tag and date range when there's more than one media file exists in the media archive
- Find the set of media files linked to a given tag and date range when there's no media file exists in the media archive
- Find the set of media files linked to a given tag and date range when the given date ranges are in future

- Find the set of media files linked to a given location and date range when the tag exists in the media archive
- Find the set of media files linked to a given location and date range when the tag doesn't exist in the media archive
- Find the set of media files linked to a given location and date range when there are no media files exist in the given date range
- Find the set of media files linked to a given location and date range when there are media files exist in the given date range

- Find the set of media files linked to a given location and date range when the given start date is before the given end date
- Find the set of media files linked to a given location and date range when the given start date is after the given end date
- Find the set of media files linked to a given location and date range when the given start date is null
- Find the set of media files linked to a given location and date range when the given end date is null
- Find the set of media files linked to a given location and date range when the given start date and the given end date are null
- Find the set of media files linked to a given location and date range when there's only one media file exists in the media archive
- Find the set of media files linked to a given location and date range when there's more than one media file exists in the media archive
- Find the set of media files linked to a given location and date range when there's no media file exists in the media archive
- Find the set of media files linked to a given location and date range when there's no media file exists in the media archive
- Find the set of media files linked to a given location and date range when the given date ranges are in future

- Find the list of media files of a given list of persons within a date range when all the persons exist in the database
- Find the list of media files of a given list of persons within a date range when only one of the persons exist in the database
- Find the list of media files of a given list of persons within a date range when none of the persons exist in the database
- Find the list of media files of a given list of persons within a date range when there are no media files exist in the given date range
- Find the list of media files of a given list of persons within a date range when there is only one media file exist in the given date range
- Find the list of media files of a given list of persons within a date range when there are many media files exist in the given date range

- Find the set of media files of a given list of persons within date range when the given start date is before the given end date
- Find the set of media files of a given list of persons within date range when the given start date is after the given end date
- Find the set of media files of a given list of persons within date range when the given start date is null
- Find the set of media files of a given list of persons within date range when the given end date is null
- Find the set of media files of a given list of persons within date range when the given start date and the given end date are null
- Find the set of media files of a given list of persons within date range when the given start date and the given end date are in the future

- Find the pictures that has all the immediate family members of a person X when there are no persons ever recorded in the database
- Find the pictures that has all the immediate family members of a person X when there is only one person ever recorded in the database
- Find the pictures that has all the immediate family members of a person X when there are many persons recorded in the database
- Find the pictures that has all the immediate family members of a person X when there are no photos ever recorded in the media archive
- Find the pictures that has all the immediate family members of a person X when there is one photo ever recorded in the media archive
- Find the pictures that has all the immediate family members of a person X when there are many photos ever recorded in the media archive
- Find the pictures that has all the immediate family members of a person X when there are no photos existing of the immediate children
- Find the pictures that has all the immediate family members of a person X when there is only photo existing for the immediate children
- Find the pictures that has all the immediate family members of a person X when there are many photos existing for the immediate children
- Find the pictures that has all the immediate family members of a person X when there are no immediate children

- Find the pictures that has all the immediate family members of a person X when there is only one immediate child
- Find the pictures that has all the immediate family members of a person X when there are many immediate children

White Box Test Cases:

Input validation (Tests on bad input data)

addPerson():

- Null passed as name
- Empty string passed as name

recordAttributes():

- Map with null attributes passed
- Map with empty attributes passed
- Null passed as person

recordRefernece():

- Null value passed as reference
- Empty string passed as reference
- Null passed as person

recordNote():

- Null value passed as reference
- Empty string passed as reference
- Null passed as person

recordPartnering():

- Null value passed as partner1
- Null passed as pertenr2

recordDissolution():

- Null value passed as partner1
- Null passed as pertenr2

addMediaFile():

- Null passed as fileLocation
- Empty string passed as fileLocation

recordMediaAttributes():

- Map with null attributes passed
- Map with empty attribute strings passed
- Null passed as fileIdentifier

peopleInMedia():

- Null passed as fileIdentifier
- Null passed as people list
- Null passed for one item in the person list

tagMedia():

- Null passed as tag
- Empty string passed as tag
- Null passed for fileIdentifier

findPerson():

- Null passed as name
- Empty string passed as name

findMediaFile():

- Null passed as name
- Empty string passed as name

findName():

- Person Id passed as Null

findMediaFile():

- Null passed as fileIdentifier field

findRelation():

- Null passed as person1
- Null passed as person2

descendents():

- Null passed as person

ancestores():

- Null passed as person

notesAndReferences():

- Null passed as person

findMediaByTag():

- Null passed as tag
- Empty string passed as tag
- Empty string passed as startDate
- Empty string passed as endDate

findMediaByLocation():

- Null passed as tag
- Empty string passed as tag
- Empty string passed as startDate
- Empty string passed as endDate

findIndividualsMedia():

- Null passed as people
- One of the items in people is null
- Empty string passed as startDate
- Empty string passed as endDate

findBiologicalFamilyMedia():

- Null passed as person

Boundary Cases (Tests at the edge of inputs)

addPerson():

- 1 character name passed

recordAttributes():

- 1 character attributes passed

- No attributes defined
- Person doesn’t exist

recordReference():

- 1 character reference passed
- No reference defined
- Person doesn’t exist

recordNote():

- 1 character note passed
- No note defined
- Person doesn’t exist

recordChild():

- No child defined
- Only 1 child defined
- Person1 doesn’t exist
- Person2 doesn’t exist

recordPartnering():

- No partner defined
- 1 partner defined
- Partner1 doesn’t exist
- Partner2 doesn’t exist

recordDissolution():

- No dissolution defined
- 1 partner defined
- Partner1 doesn’t exist
- Partner2 doesn’t exist

addMediaFile():

- 1 character filename passed

recordMediaAttributes():

- 1 character attributes passed
- No attributes defined

- fileIdentifier doesn’t exist

peopleInMedia():

- 0 persons in media
- 1 person in media
- Fileidentifier doesn’t exist
- One of the personIdentity in people doesn’t exist

tagMedia():

- 0 tags defined
- 1 tag defined
- FileIdentifier doesn’t exist

findPerson():

- No persons defined
- Person doesn’t exist
- 1 person defined

findMediaFile():

- No media file defined
- Media doesn’t exist
- 1 media file defined

findName():

- No persons defined
- Person doesn’t exist
- 1 person defined

findMediaFile():

- No media file defined
- Media doesn’t exist
- 1 media file defined

findRelation():

- No persons defined
- Person1 doesn’t exist
- Person2 doesn’t exist

- 2 persons defined

descendents():

- 0 descendents defined
- Person doesn’t exist
- 1 descendent defined

ancestores():

- 0 ancestores defined
- Person doesn’t exist
- 1 anscestor defined

notesAndReferences():

- 0 notes and 0 references defined
- 1 note and 1 reference defined
- Person doesn’t exist

findMediaByTag():

- 0 tags in the given range
- 1 tag in the given range
- Both dates are null

findMediaByLocation():

- 0 media in the given location
- 1 media in the given location
- Both dates are null

findIndividualsMedia():

- 0 individuals in the given range
- 1 individual in the given range
- One of the personIdentity in people doesn’t exist

findBiologicalFamilyMedia():

- Person doesn’t exist
- 0 children for the given person
- 1 child for the given person with 0 media defined
- 1 child for the given person with 1 media defined

Control Flow Cases (Tests of the core operations)

addPerson():

- Duplicate name passed
- Add a person when there are no persons defined
- Add a person when there is one person defined
- Add a person when there are many persons defined

recordAttributes():

- Duplicate attributes passed
- All attributes have only one line in them
- Add attributes when there 0 attributes linked to that person
- Add attributes when there 1 attribute linked to that person
- Add attributes when there are many attributes linked to that person

recordReference():

- Duplicate attributes passed
- All references have only one line in them
- Add a reference when there is 0 reference linked to the person
- Add a reference when there is 1 reference linked to the person
- Add a reference when there are multiple references linked to the person

recordNote():

- Duplicate note passed
- All notes have only one line in them
- Add a note when there is 0 note linked to the person
- Add a note when there is 1 note linked to the person
- Add a note when there are multiple notes linked to the person

recordChild():

- Duplicate child passed
- Parent and child exists
- Many children already recorded
- Add a child when there are no children linked to the person
- Add a child when there is 1 child linked to the person
- Add a child when there are many children linked to the person

- Add a parent to the child when there is no parent defined previously
- Add a parent to the child when there is 1 parent defined previously
- Add a parent to the child when there are multiple parents defined previously.

recordPartnering():

- Duplicate partnering relation passed
- Partner1 and Partner2 exists
- Record partnering when there are no previous relations defined
- Record partnering when there is the previous relation already defined

recordDissolution():

- Duplicate dissolution passed
- Partner1 and Partner2 exists
- Record dissolution when there are no previous relations defined
- Record dissolution when there is one previous relation defined

addMediaFile():

- Duplicate media file
- Add a media file when there are no media files in the media archive
- Add a media file when there is 1 media file in the media archive
- Add a media file when there are many media files existing in the media archive

recordMediaAttributes():

- Duplicate media attributes
- 1 line attributes
- Record media attributes to a media file with 0 attributes added before
- Record media attributes to a media file with 1 attribute added
- Record media attributes to a media file with many attributes already linked

peopleInMedia():

- Duplicate people in the media
- Many people in the media

- Media file exists

tagMedia():

- 1 line tags in the media
- Duplicate tags in the media
- Add another tag when already there is one tag linked
- Add another tag when there are multiple tags already linked

findPerson():

- Find a person when there are many persons with the same name
- Find a person when there is no one with the given name

findMediaFile():

- Find a media file when there are many files with the same name
- Find a media file when there is no file with the given name

findName():

- Find a person's name when there are no persons with the given id
- Person exists

findMediaFile():

- Find a media file when there are no persons with the given id
- Media file exists

findRelation():

- There is 1 relation between two partners
- Both persons exist
- Find relation when the persons are partners
- Find relation when the persons are divorced
- Find the relation when there is no common ancestor
- Find the relation when there is 1 common ancestor
- Find the relation when there are many common ancestors
- Find the relation when the degree of removal is 0
- Find the relation when nA is 0, nB is 0
- Find the relation when min{nA, nB} is 1

nA is the number of generations A is back from the common ancestor

nB is the number of generations B is back from the common ancestor

descendents():

- Find descendants when there is 1 descendant lined to the given person
- Find descendants at generation 0
- Find descendants by giving a negative generation

ancestores():

- Find ancestores when there is 1 ancestore lined to the given person
- Find ancestores at generation 0
- Find ancestores by giving a negative generation

notesAndReferences():

- Person exists
- Find notes and references when there are many notes and references defined

findMediaByTag():

- Tag exists
- Find tags of a media file when there is one tag linked to the media file
- Find tags of a media file when there are more tags linked to the media file

findMediaByLocation():

- Location exists
- Find media file when there is one location specified to the media file
- Find media file when there are no media files linked to the given location
- Find media file when there are many media files linked to the given location

findIndividualsMedia():

- Set of persons exist
- Find media file when there is one person specified to the media file
- Find media file when there are many media files linked to the given person

findBiologicalFamilyMedia():

- There is a child to the given person
- There are many media files that exist to the given person's immediate child

Data Flow Cases (Tests around the order in which things are done)

recordeAttributes():

- Record the attributes before adding the person to the database
- Record the attributes after adding the person to the database

recordReference():

- Record reference before adding the person to the database
- Record reference after adding the person to the database

recordNote():

- Record note before adding the person to the database
- Record note after adding the person to the database

recordChild():

- Record child before adding the parent to the family tree database
- Record child after adding the parent to the family tree database
- Record the child to another parent before added to the family tree database
- Record the child to another parent after recording dissolution for both the parents

recordPartnering():

- Record partnering before adding the persons
- Record partnering after dissolution between the same persons

recordDissolution():

- Record dissolution before partnering them
- Record dissolution after partnering them

recordMediaAttributes():

- Record the attributes before adding the file to the media archive
- Record the attributes after adding the file to the media archive

peopleInMedia():

- Add people before adding the file to the media archive
- Record the people after adding the file to the media archive

tagMedia():

- Add tags before adding the file to the media archive
- Record the tags after adding the file to the media archive

findPerson():

- Find the person before adding the person to the family tree

findMediaFile():

- Find the media file before adding to the media archive

findName():

- Find the person before adding the person to the family tree

findMediaFile():

- Find the media file before adding to the media archive

findRelation():

- Find relation of two persons before defining a relation between them
- Find the relation between two persons before adding any of the person to the family tree.

decendants():

- Find descendants before adding any person to the family tree
- Find descendants before making any relations to the persons

ancestores():

- Find ancestors before adding any person to the family tree
- Find ancestors before making any relations to the persons

notesAndReferences():

- Find notes and references before adding any peron to the family tree
- Find notes and references before recording notes and references to the persons

findMediaByTag():

- Find tags of a media before adding any media to the media archive
- Find tags of a media file before recording any tags to the media

findMediaByLocation():

- Find Location of a media before adding any media to the media archive
- Find Location of a media file before recording any location to the media

findIndividualsMedia():

- Find Individuals of a media before adding any media to the media archive
- Find Individuals of a media file before linking any people to the media file.

findBiologialMedia():

- Find media before adding any persons to the family tree
- Find media before adding a child to the person
- Find media before adding any media to the media archive