CSCI 3901

Software Development Concepts

Final Project
Test Plan

Prepared By

Bhavisha Oza (B00935827)

Test Plan Index:

1. Introduction

The purpose of this test plan is to outline the testing approach for the PublicationLibrary class that is being developed as part of the CSCI 3901 Final Project. The testing approach will ensure that the class meets the functional requirements and that it is robust, reliable, and performs as expected.

The plan identifies the things that need to be tested, the features that need to be tested, the sorts of tests that need to be done, the people in charge of doing the tests, the materials and time needed to finish the tests, and the risks connected to the plan.

1.1.1 In Scope: All the features mentioned below are need to be tested:

- Add publication
- Add references
- Add venue
- Add publisher
- Add area
- Get publications
- Add Author citations
- Seminal papers
- Collaborators
- authorRessearch area
- convert citations in a paper into actual IEEE references.

1.1.2 Out of Scope: The given features are not tochecked on any User Interfaces as it is mentioned in requirement that “This project is _not_ expected to include a user interface.”

2. Test Methodology

The testing will be done through a combination of manual and automated testing. Unit testing will be done for each method of the PublicationLibrary class, and integration testing will be done to ensure that the methods work together seamlessly. The testing will be done using a combination of positive and negative test cases.

2.1 Types of Testing:

- Unit testing
- Integration testing
- Functional testing
- Non-functional testing
- Smoke testing
- Bug Regression testing
- Iteration/Regression testing
- User Acceptance testing

2.2 Test Coverage:

The test coverage for the PublicationLibrary class will include the following aspects:

- Method functionality
- Exception handling
- Error messages
- Data insertion

2.3 Test Strategy:

Identify Entry level and Exit level criteria

3. Test Deliverables

Test deliverables are provided as below:

Before testing phase:

- Test plan document.
- Test cases documents.

During the testing:

- Test Data
- Bug ids

After the testing cycles:

- Test Results/reports
- Defect Report

3.1 Test cases:

boolean addPublication (String identifier, Map<String, String> publicationInformation)

input validations

- Verify when trying to add a publication with an empty or null identifier.
- Verify when trying to add a publication with an identifier that exceeds the maximum length allowed.
- Verify when trying to add a publication with an empty or null value for any of the mandatory information fields, such as author, title, or publication venue.
- Verify when trying to add a publication with invalid values for any of the mandatory or optional information fields, such as a non-numeric value for the volume or issue number.

boundary case

- Verify when trying to add a publication with an identifier that exceeds the maximum length allowed by the system.
- Verify when trying to add a publication with an empty or null value for any of the mandatory information fields, such as author, title, or publication venue.
- Verify that the method properly handles edge cases for the publication information fields, such as the minimum and maximum length allowed for the author names, or the maximum number of authors allowed for a publication.
- Verify when the publication information fields are at the minimum.
- Verify when the publication information fields are at the maximum.

Control flow case

- Verify that a publication with valid inputs is added successfully to the library and the method returns true.
- Verify that the method returns false when trying to add a publication with an invalid or missing identifier.
- Verify that the method returns false when trying to add a publication with incomplete or missing publication information.
- Verify that the method returns false when trying to add a publication with a duplicate identifier.

data flow
- Call when publication was successfully added.
- Call when publication was not successfully added.
- Call when there are multiple authors listed in the author field, and correctly parses and stores all of the author names.

boolean addReferences (String identifier, Set<String > references)

input validations
- Verify when trying to add a publication with an empty or null identifier.
- Verify when trying to add a publication with an identifier that exceeds the maximum length allowed.
- Verify when trying to add a publication with an empty or null value for any of the mandatory information fields, such as author, title, or publication venue.
- Verify when trying to add a publication with invalid values for any of the mandatory or optional information fields, such as a non-numeric value for the volume or issue number.

boundary case
- When identifier is null/empty.
- Reference is null/empty.
- Reference is duplicate.
- Reference has invalid character.

Control flow case
- adding references for a new publication identifier.
- adding references for an existing publication identifier.
- adding duplicate references for a publication identifier
- adding invalid references for a publication identifier.
- adding references to a publication identifier that doesn't exist.

data flow
- Call when publication with an empty reference list before the function call.
- Call when publication with the updated reference list after the function call.

boolean addVenue (String venueName, Map<String, String> venueInformation,

Set<String> researchAreas)

input validations

- Verify when the venueName parameter is null or empty.
- Verify when the venueInformation parameter is null or empty.
- Verify when the researchAreas parameter is null or empty.
- Verify when the venueInformation parameter does not contain mandatory information such as publisher, location, or conference_year.
- Verify when the venueInformation parameter contains invalid data.
- Verify when the researchAreas parameter contains invalid data.
- Verify when the venueName parameter already exists in the library, to prevent adding duplicate venue names.

boundary case
- When identifier is null/empty.
- Reference is null/empty.
- Reference is duplicate.
- Reference has invalid character.

Control flow case
- Verify when a new venue is successfully added to the library with all mandatory information.
- Verify when a new venue is not added to the library due to missing mandatory information.
- Verify that the method can add a venue with all the possible information (publisher, editor, editor_contact, location, and conference_year).
- Verify able to add multiple venues with different names and information.
- Verify adding venues with the same name but different information.

Dataflow cases
- Verify that the existing venues in the library are not affected by adding a new venue.
- Verify that the method can still add a new venue after the library has been closed and reopened.
- Verify that the method can still add a new venue after the library has been updated to a newer version.

boolean addPublisher (String identifier, Map<String, String> publisherInformation)

input validations
- Verify when attempting to add a publisher with a non-string identifier
- Verify when attempting to add a publisher with a missing or empty identifier
- Verify when attempting to add a publisher with a missing or invalid contact_name
- Verify when attempting to add a publisher with a missing or invalid contact_email
- Verify when attempting to add a publisher with a missing or invalid location

boundary case
- Verify when a very large number of publishers being added to the library.
- Verify when the minimum possible information
- Verify when a publisher with the maximum possible information (

Control flow cases

- Add a new publisher to the library with valid identifier and publisher information.
- Attempt to add a publisher with an identifier that already exists in the library.
- Attempt to add a publisher with missing required information.
- Add a publisher with all possible information.

Data flow cases

- Attempt to add a publisher after the library has already been initialized with publishers. Verify that existing publishers are not affected and that new publishers can still be added.
- Remove a publisher from the library and then attempt to add it again with the same identifier. Verify that the method returns true, and the publisher is added back to the library.

boolean addArea (String researchArea, Set<String> parentArea)

input validations

- Verify if the function can handle invalid research area name.
- Verify if the function can handle invalid parent area names.

boundary case

- Verify if the function can handle adding the same research area multiple times.
- Verify if the function can handle adding the maximum allowed number of research areas.
- Verify if the function can handle adding a research area with the maximum allowed length.
- Verify if the function can handle adding a research area with a parent area set that is the maximum allowed length.

Control flow cases

- Verify if a research area can be added with valid inputs.
- Verify if a research area can be added without parent area.
- Verify if a research area can be added with a non-existing parent area.
- Verify if a research area can be added with an empty research area name.

Data flow cases

- Verify if the performance of the function is efficient when adding multiple research areas

Map<String, String> getPublications (String key)

input validations

- Key is null/empty
- non-existing publication key is passed as input.

boundary case

- when the publication key is at the minimum valid value.
- when the publication key is at the max valid value..

control flow cases

- Verify if the method returns a map of all the information for a publication when a valid publication key is passed as input.
- Verify if the method returns null when an invalid publication key is passed as input.

Data flow cases
- Verify the response time of the method for different input keys.

int authorCitations (String author)

input validations
- Author is null/empty
- Author name contains special characters.

boundary case
- an author name with no citations is provided.

Control flow cases
- Test that the method returns 0 if the given author has not been cited in any publication.
- Test that the method returns the correct number of citations for a given author who has been cited in multiple publications.
- Test that the method can handle non-ASCII characters in the author's name

Data flow cases
- Test that the method still works after making changes to the PublicationLibrary class or the data it stores.

Set<String> seminalPapers (String area, int paperCitation, int otherCitations)

input validations
- Area, paperCitation, otherCitation is null/empty
- area name contains special characters.

boundary case
- an author name with no citations is provided.

Control flow cases
- Test that the method returns 0 if the given author has not been cited in any publication.
- Test that the method returns the correct number of citations for a given author who has been cited in multiple publications.
- Test that the method can handle non-ASCII characters in the author's name

Data flow cases
- Verify that the function can handle a large number of papers in the given area and the citation data.
- Verify that the function returns the same set of seminal papers for the same input values, even if called multiple times.

Set<String> collaborators (String author, int distance)

input validations
- Author, distance is null/empty
- author name contains special characters.
- Distance is negative

boundary case
- when the distance parameter is equal to the maximum author distance in the publication library.
- when there is only one author in the publication library.

Control flow cases
- Verify when the given author doesn't exist in the publication library.
- Verify when the distance parameter is less than 1.
- Verify that the method returns the correct set of collaborators for a given author and distance.
- Verify that the method handles cyclic author relationships correctly.
- Verify when there are multiple paths between two authors.

DataFlow cases
- Verify that the method correctly retrieves the author and publication information from the PublicationLibrary class.
- Verify that the method correctly calculates the author distance between two authors by traversing the co-authorship links.

Set<String> authorResearchAreas (String author, int threshold)

input validations
- when author's name is not provided.
- when the threshold is not provided.
- when the threshold is not a positive integer.
- when the author's name contains invalid characters.
- when the author's name is too long.

boundary case
- Verify the smallest possible threshold value.
- Verify the largest possible threshold value.
- Verify the author's name with the smallest possible length.
- Verify the author's name with the largest possible length.
- Verify the maximum number of research areas.

Control flow cases
- Verify when the author has not published any papers.
- Verify when the author's name is not in the database.
- Verify when the threshold is set to 1.
- Verify when the threshold is set to a value greater than 1.

- Verify when a research area is part of a larger research area.
- Verify if the author's name is provided in a different case.

Data flow cases
- Verify that the function returns the same output when given the same input, even after changes have been made to the code.
- Verify that the function still works as expected after changes have been made to the database or other parts of the system.
- Verify that the function does not affect the functionality of other parts of the system.

Paper conversion

input validations
- when the input files don't not exist or are not in the correct format.

boundary case
- when a citation key is the first or last item in the citation list.
- when a citation list contains the maximum number of citations allowed by the IEEE format.

Control flow cases
- when a single citation converted into an IEEE formatted reference in the output text.
- when multiple citations converted into a list of IEEE formatted references in the output text.
- when a citation key does not exist in the publication library.
- when the IEEE formatted reference appends properly to the end of the paper with the citation key in square brackets.
- when the input file contains no citations.
- when the output file already exists and appends the converted references to the end of the file

Data flow cases
- Check the performance by providing an input file with a large number of citations and ensuring that the method runs efficiently.
- Check by providing an input file with a large amount of text and citations and ensuring that the method can handle it without crashing or encountering errors.

4. Environment Needs

- Java 11 or later
- IDE: preferably IntelliJ IDEA CE
- Programming Language: Java
- Git version control
- Testing Framework: JUnit 5