

Monash University: Assessment Cover Sheet

| | | | |
|--------------------------------|---|--|-------------------|
| Student name | Karkera | Aishwarya Vasant | |
| School/Campus | Caulfield | Student's I.D. number | 29973740 |
| Unit name | FIT5171 System validation and verification, quality and standards S1 2020 | | |
| Lecturer's name | Yuan-Fang Li | Tutor's name | Ningran (Icey) Li |
| Assignment name | Assignment 1 submission | Group Assignment: Yes Note, each student must attach a coversheet | |
| Lab/Tute Class: Tuesday | Lab/Tute Time: 4pm-6pm | Word Count: 2566 | |
| Due date: 19-04-2020 | Submit Date: 19/04/2020 | Extension granted <input type="checkbox"/> | |

If an extension of work is granted, specify date and provide the signature of the lecturer/tutor. Alternatively, attach an email printout or handwritten and signed notice from your lecturer/tutor verifying an extension has been granted.

Extension granted until (date):/...../..... Signature of lecturer/tutor:

| Late submissions policy | Days late | Penalty applied |
|--|-----------|-----------------|
| Penalties apply to late submissions and may vary between faculties. Please refer to your faculty's late assessment policy for details. | | |

Patient/client confidentiality: Where a patient/client case study is undertaken a signed [Consent Form](#) must be obtained.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means to take and use another person's ideas and or manner of expressing them and to pass these off as one's own by failing to give appropriate acknowledgement. This includes material from any source, staff, students or the Internet - published and unpublished works.

Collusion: Collusion means unauthorised collaboration on assessable written, oral or practical work with another person. Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or nominee, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity [Policy](#) and [Procedures](#)
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) [Regulations](#) (academic misconduct).
- I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may, for the purposes of assessment, reproduce the assignment and:
 - i. provide it to another member of faculty and any external marker; and/or
 - ii. submit to a text matching/originality checking software; and/or
 - iii. submit it to a text matching/originality checking software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration or otherwise breached the academic integrity requirements in the Student Academic Integrity [Policy](#).

Date: 19/04/2020 Signature: Akarkera *

Privacy Statement:

For information about how the University deals with your personal information go to <http://privacy.monash.edu.au/guidelines/collection-personal-information.html#enrol>

Monash University: Assessment Cover Sheet

| | | | |
|------------------------|---|--|-------------------|
| Student name | Jere | Neha Nanda | |
| School/Campus | Caulfield | Student's I.D. number | 29994985 |
| Unit name | FIT5171 System validation and verification, quality and standards S1 2020 | | |
| Lecturer's name | Yuan-Fang Li | Tutor's name | Ningran (Icey) Li |
| Assignment name | Assignment 1 submission | Group Assignment: Yes Note, each student must attach a coversheet | |
| Lab/Tute Class: | Tuesday | Lab/Tute Time: | 4pm-6pm |
| Due date: | 19-04-2020 | Submit Date: | 19-04-2020 |
| | | Word Count: | 2566 |
| | | Extension granted <input type="checkbox"/> | |

If an extension of work is granted, specify date and provide the signature of the lecturer/tutor. Alternatively, attach an email printout or handwritten and signed notice from your lecturer/tutor verifying an extension has been granted.

Extension granted until (date):/...../..... Signature of lecturer/tutor:

| Late submissions policy | Days late | Penalty applied |
|--|-----------|-----------------|
| Penalties apply to late submissions and may vary between faculties. Please refer to your faculty's late assessment policy for details. | | |

Patient/client confidentiality: Where a patient/client case study is undertaken a signed [Consent Form](#) must be obtained.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means to take and use another person's ideas and or manner of expressing them and to pass these off as one's own by failing to give appropriate acknowledgement. This includes material from any source, staff, students or the Internet - published and unpublished works.

Collusion: Collusion means unauthorised collaboration on assessable written, oral or practical work with another person. Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or nominee, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity [Policy](#) and [Procedures](#)
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) [Regulations](#) (academic misconduct).
- I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may, for the purposes of assessment, reproduce the assignment and:
 - i. provide it to another member of faculty and any external marker; and/or
 - ii. submit to a text matching/originality checking software; and/or
 - iii. submit it to a text matching/originality checking software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration or otherwise breached the academic integrity requirements in the Student Academic Integrity [Policy](#).

Date:/04/.....2020 Signature:..... Neha Jere..... *

Privacy Statement:

For information about how the University deals with your personal information go to <http://privacy.monash.edu.au/guidelines/collection-personal-information.html#enrol>

Monash University: Assessment Cover Sheet

| | | | |
|--------------------------------|---|--|-------------------|
| Student name | Rezwan | Farhad Ullah | |
| School/Campus | Caulfield | Student's I.D. number | 30270111 |
| Unit name | FIT5171 System validation and verification, quality and standards S1 2020 | | |
| Lecturer's name | Yuan-Fang Li | Tutor's name | Ningran (Icey) Li |
| Assignment name | Assignment 1 submission | Group Assignment: Yes Note, each student must attach a coversheet | |
| Lab/Tute Class: Tuesday | Lab/Tute Time: 4pm-6pm | Word Count: 2566 | |
| Due date: 19-04-2020 | Submit Date: 19-04-2020 | Extension granted <input type="checkbox"/> | |

If an extension of work is granted, specify date and provide the signature of the lecturer/tutor. Alternatively, attach an email printout or handwritten and signed notice from your lecturer/tutor verifying an extension has been granted.

Extension granted until (date):/...../..... Signature of lecturer/tutor:

| Late submissions policy | Days late | Penalty applied |
|--|-----------|-----------------|
| Penalties apply to late submissions and may vary between faculties. Please refer to your faculty's late assessment policy for details. | | |

Patient/client confidentiality: Where a patient/client case study is undertaken a signed [Consent Form](#) must be obtained.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means to take and use another person's ideas and or manner of expressing them and to pass these off as one's own by failing to give appropriate acknowledgement. This includes material from any source, staff, students or the Internet - published and unpublished works.

Collusion: Collusion means unauthorised collaboration on assessable written, oral or practical work with another person. Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or nominee, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity [Policy](#) and [Procedures](#)
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) [Regulations](#) (academic misconduct).
- I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may, for the purposes of assessment, reproduce the assignment and:
 - i. provide it to another member of faculty and any external marker; and/or
 - ii. submit to a text matching/originality checking software; and/or
 - iii. submit it to a text matching/originality checking software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration or otherwise breached the academic integrity requirements in the Student Academic Integrity [Policy](#).

Date:19/04/2020..... Signature:..... Farhad Ullah Rezwan *

Privacy Statement:

For information about how the University deals with your personal information go to <http://privacy.monash.edu.au/guidelines/collection-personal-information.html#enrol>

Monash University: Assessment Cover Sheet

| | | | |
|------------------------|---|------------------------------|--|
| Student name | Kang | Zhibin | |
| School/Campus | Caulfield | Student's I.D. number | 29600189 |
| Unit name | FIT5171 System validation and verification, quality and standards S1 2020 | | |
| Lecturer's name | Yuan-Fang Li | Tutor's name | Ningran (Icay) Li |
| Assignment name | Assignment 1 submission | | Group Assignment: Yes Note, each student must attach a coversheet |
| Lab/Tute Class: | Tuesday | Lab/Tute Time: | 4pm-6pm |
| Word Count: | 2566 | | |
| Due date: | 19-04-2020 | Submit Date: | 19-04-2020 |
| | | Extension granted | <input type="checkbox"/> |

If an extension of work is granted, specify date and provide the signature of the lecturer/tutor. Alternatively, attach an email printout or handwritten and signed notice from your lecturer/tutor verifying an extension has been granted.

Extension granted until (date):/...../..... Signature of lecturer/tutor:

| Late submissions policy | Days late | Penalty applied |
|--|-----------|-----------------|
| Penalties apply to late submissions and may vary between faculties. Please refer to your faculty's late assessment policy for details. | | |

Patient/client confidentiality: Where a patient/client case study is undertaken a signed [Consent Form](#) must be obtained.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means to take and use another person's ideas and or manner of expressing them and to pass these off as one's own by failing to give appropriate acknowledgement. This includes material from any source, staff, students or the Internet - published and unpublished works.

Collusion: Collusion means unauthorised collaboration on assessable written, oral or practical work with another person. Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or nominee, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity [Policy](#) and [Procedures](#)
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) [Regulations](#) (academic misconduct).
- I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may, for the purposes of assessment, reproduce the assignment and:
 - i. provide it to another member of faculty and any external marker; and/or
 - ii. submit to a text matching/originality checking software; and/or
 - iii. submit it to a text matching/originality checking software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration or otherwise breached the academic integrity requirements in the Student Academic Integrity [Policy](#).

Date: 19 / 04 / 2020 Signature: Zhibin Kang *

Privacy Statement:

For information about how the University deals with your personal information go to <http://privacy.monash.edu.au/guidelines/collection-personal-information.html#enrol>



Test Planning, System Setup and Code Understanding & Extension

FIT5171 ASSIGNMENT 1

Submitted By:

Neha Jere

29994985

Aishwarya Karkera

29973740

Farhad Ullah Rezwan

30270111

Zhibin Kang

29600189

Table of Contents:

| | |
|---|---|
| 1. Testing Strategy..... | 2 |
| 1.1. Scope and overview: | 2 |
| 1.2. Types of testing:..... | 2 |
| 1.3. Testing tools and approach:..... | 2 |
| 1.4. Defect Tracking Mechanisms: | 3 |
| 2. Development and Testing Environment Setup: | 3 |
| 2.1. Local Devices Setup: | 3 |
| 2.2. Remote Setup: | 5 |
| 3. Code based understanding and extension..... | 5 |
| Contribution Declaration Form..... | 6 |

1. Testing Strategy

1.1. Scope and overview:

This report gives an overview of the testing approaches which have been followed by the team members to build the web application, www.allaboutecm.com. It also outlines the types of testing which have been used to test different test cases of the web application to allow users to keep track of their ECM records and discover new artists. Reporting the bugs or errors using a defect tracking tool has also been demonstrated in the strategy.

1.2. Types of testing:

Behaviour based testing/Functional Testing:

1. Equivalence class partitioning:

Equivalence class partitioning is used to decrease the number of potential test cases that are required to test a system by grouping all the possible inputs which are entered by the user into equal classes. The range of inputs entered by the users work similarly throughout the system due to the equivalent classes and considering the set of conditions in each partition is the same. This technique is used to reduce the number of test cases while allowing a larger test coverage of the system. The number of test cases are reduced because testing is done only on one condition from each partition. For example: The Input of the album release year should be a 4-digit number between 1970 and current year

2. Boundary Value tests:

Using boundary value analysis, the number of issues or errors found on the defined input values at the boundaries between the partitions are higher. The boundary of the partition is the value near the limit where the behaviour of the system changes. The valid and invalid inputs both are being tested to verify the issues. The system works well for all the values between the two boundary values if these two values have been tested and makes the system perform well. For example: `releaseYearShouldBeBetween1970AndCurrent`

3. Decision based Tests:

Using decision table testing technique, different input combinations are used to test the systems behaviour. This technique is used to ensure good test coverage of a system where for each set of input values the system behaviour is different as opposed to a range of inputs which are used in BVT and ECP. For example: `recordNumberCanOnlyAcceptAlphanumericWithSpaceORWithForwardSlash`

Non-Functional testing:

1. Load tests:

In the load testing process, developers are going to simulate the expected number of users using the ECM record system to verify expected response times. Thus, developers can get how many users this system can handle. We expect the response time should lower than 3 seconds.

2. Stress tests:

When the number of users is much more than we expected, because of some events, developers should test whether the ECM system, hardware are going to crash. Meanwhile, we also need to test how long the ECM system takes to return to normal operational levels after the events.

3. Usability tests:

Usability tests are designed to verify if the software product can be easily used by end-users. End-users should use this system easily. Make sure users do not have confusion on functions like searching albums, tracking artists, etc.

4. Maintainability tests:

Comments should be added in source code and test code. Meanwhile, developers should write `@Displayname` on test cases to simplify maintainability.

1.3. Testing tools and approach:

The web application project of www.allaboutecm.com is continuously tested with automatic testing, where a well written test plan is followed. The benefit of automatic testing is it makes regression testing more convenient, especially

when the program is modified more frequently. Thus, this approach is suitable for TDD. Meanwhile the scripts and use cases for the tests are well designed and the assertions results of the tests are predictable, automating testing can greatly improve efficiency and reduce testing time.

The functional requirement for the users from this website is that it can showcase, update or add, and analyse album related information that is already existed in ecmrecords.com website. Developers conducted a proper test plan and testing strategy for the purpose of automatic testing. Automatic testing tools that are used for this web application is Junit. Junit is an open-source testing framework for Java8 and above. Junit provides functions of organizing tests and running tests. It allows developers to write assertions on expected results of execution and directly run to get results. According to the results returned, developers can go back to the source code part to fix the fail part.

1.4. Defect Tracking Mechanisms:

Maven is configured to generate reports about the errors in order to track the failed test cases. A plugin incorporated in maven is used to generate unit tests of the application. The surefire plugin is utilized while running the test cases of the build and generates reports to provide the developer with analysis of any defects or errors. This makes it easy for the developer to identify the error and fix it to improve the performance of the application. The bug tracker in the system keeps track of the software errors and also maintains a list of backlogs which helps in improving the development management of the software or application. A specific individual from the development team keeps track of the bugs using the defect tracking tool which is the backlog to analyse the report which is generated once the bug is identified. This efficient tool makes it easy for the development team to resolve defects and update the necessary defect using the defect tracking tool.

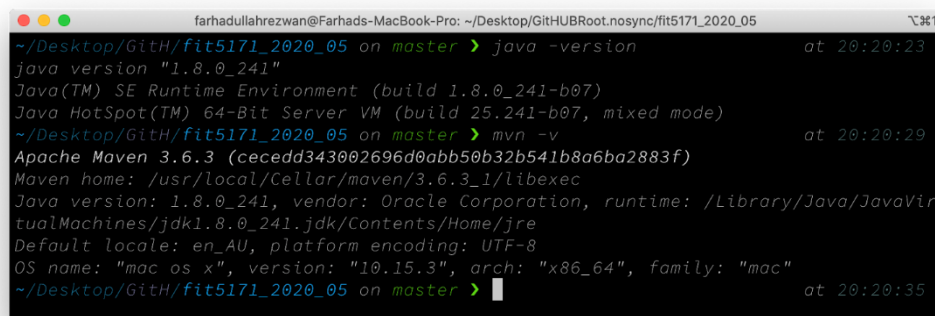
2. Development and Testing Environment Setup:

The development environment used for this project is IntelliJ IDEA IDE with JAVA JDK version of 8(1.8) and Apache Maven version of 3.6. The version control system used for this team project is GitHub, with a corresponding education pack enabled for each team member. Some team members used 3rd party GitHub software called GitKraken and others used the command line for continuously integrating the development and testing changes to the remote git repository.

2.1. Local Devices Setup:

Java JDK: Java JDK version 8(1.8) is used for the testing and development environment for the Web Application allaboutecm.com (Figure 2.1). JDK 8 is highly compatible with build management tool Apache Maven, testing framework JUNIT, and other frameworks like Spark, Neo4j and Freemarker.

Apache Maven: Build automation, project dependency and documentation tool used for the development of this web application is Apache Maven (version 3.6 and above), as shown in Figure 2.1.

A screenshot of a terminal window on a Mac. The window title is 'farhadullahrezwan@Farhads-MacBook-Pro: ~/Desktop/GitHUBRoot.nosync/fit5171_2020_05'. The terminal shows the following commands and output:

```
~/Desktop/GitH/fit5171_2020_05 on master > java -version at 20:20:23
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)

~/Desktop/GitH/fit5171_2020_05 on master > mvn -v at 20:20:29
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: /usr/local/Cellar/maven/3.6.3_1/libexec
Java version: 1.8.0_241, vendor: Oracle Corporation, runtime: /Library/Java/JavaVirtualMachines/jdk1.8.0_241.jdk/Contents/Home/jre
Default locale: en_AU, platform encoding: UTF-8
OS name: "mac os x", version: "10.15.3", arch: "x86_64", family: "mac"

~/Desktop/GitH/fit5171_2020_05 on master > at 20:20:35
```

Figure 2.1: Local Device setup for JDK and Apache Maven

IntelliJ IDEA: The integrated development environment (IDE) used for the web application is IntelliJ IDEA Ultimate (figure 2.3). Powerful features with this IDE that helped the testing and development of allaboutecm.com web application project smoothly is smart completion, inline debugging and build tool integration ie. Maven.

Making sure IDE works with Apache Maven: The built-in Maven support (version 3.1.x) for IntelliJ IDEA Ultimate has some issues with the compatibility with maven-surefire-report-plugin (figure 2.2). This issue for the local devices is

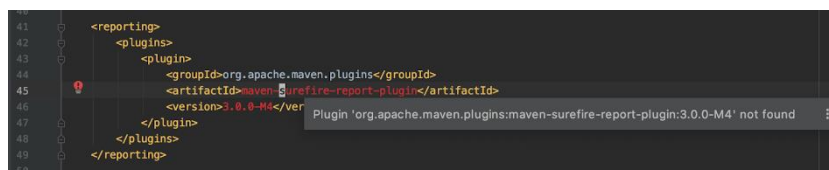


Figure 2.2: maven-surefire-report-plugin compatibility issue with IntelliJ IDEA with Default dependency tool Maven 3.1.

solved by defining the directory for the Maven (version 3.6) in the local environment, and as a result we can utilize the surefire plugin we can run all the test cases with a single click (figure 2.3)

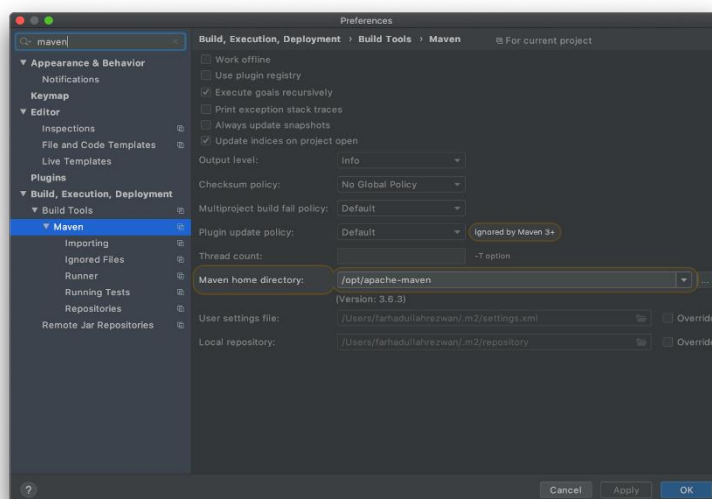
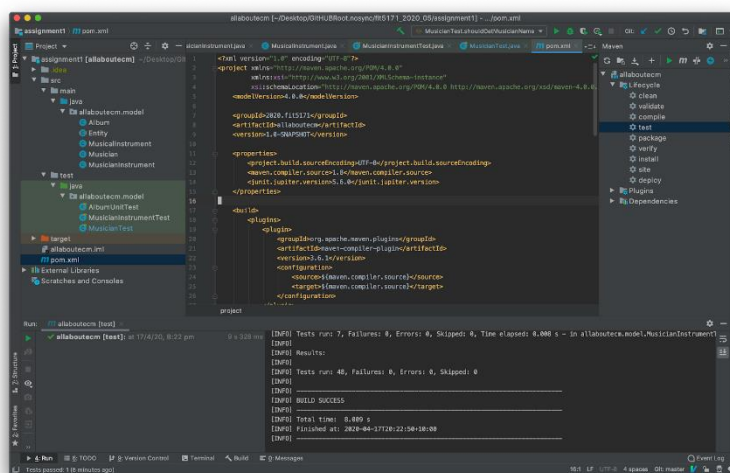


Figure 2.3: Integrated Development Environment IDE), IntelliJ IDEA, with custom directory for Apache Maven (version 3.6)

GitHub private repository and local device setup: GitHub Education pack is used to create the private repository for the project. After cloning the repository in the local respective devices, team members used GitKraken and Command Line to pull, commit, and push changes, and creating, merging feature branches with the master branch.

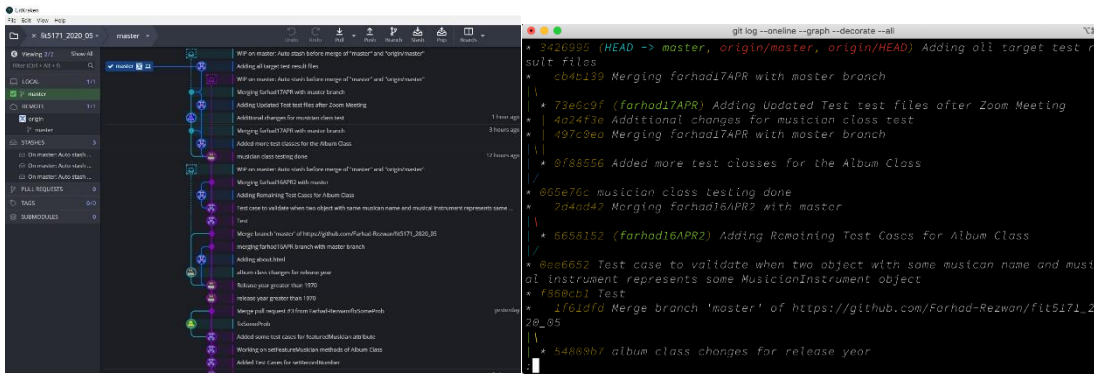


Figure 2.4: Git operations in local devices.

2.2. Remote Setup:

For the allboutecm.com web application popular version control tool GitHub is used. Updates and commits for the team source code is perfectly portrayed in the remote repository GitHub as shown in the below figure 2.5.

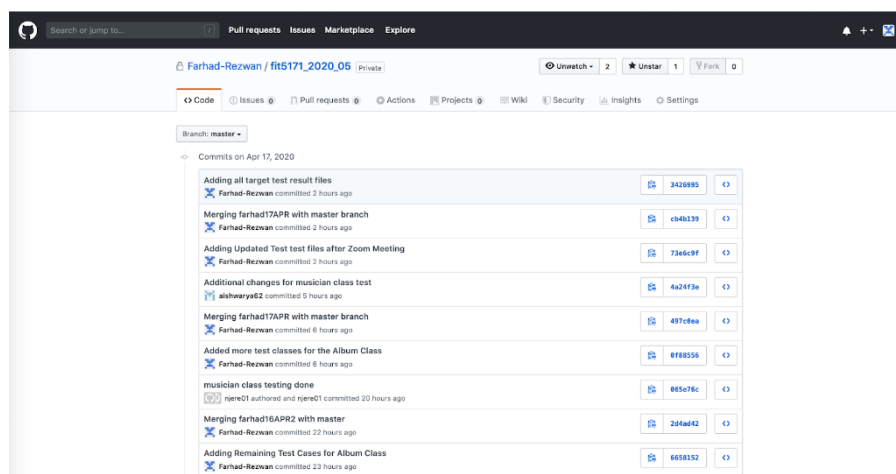


Figure 2.5: Remote GitHub Repository, portraying the local device commits.

3. Code based understanding and extension

The test cases written below have been successfully tested and have been passed. For all the test cases that check null and blank conditions, we have extended the respective classes and attributes of the classes in their setters by appropriate annotations in their main class and then tested the test case by adding the assertions in the test class.

ALBUMUNITTEST

1. Constraint: Record Number and Album Name should uniquely identify each album.

Unit Tests and their respective methods:

- Record Number Cannot be Null
➔ `shouldThrowExceptionWhenRecordNumberSetToNull`
- Album Name Cannot be Null
➔ `albumNameCannotBeNull`
- Record Number cannot be empty or blank
➔
- Album Name cannot be empty or blank
➔ `shouldThrowExceptionWhenAlbumNameSetToNull`

- e. Same record number and album name should represent one unique Album object
→ `sameNameAndNumberMeansSameAlbum`
- f. Album URL cannot be null
→ `shouldThrowExceptionWhenAlbumURLSetToNull`
- g. Track cannot be null
→ `shouldThrowExceptionWhenTracksSetToNull`

2. Constraint: Release Year has to be between 1970 to current.

Unit tests and their respective methods:

- a. Release Year Should be between 1970 and the current year
→ `releaseYearShouldBeBetween1970AndCurrent`
[For this test case we extended the set method of release year of Album class with the below code:

```
int year = Calendar.getInstance().get(Calendar.YEAR);
if((releaseYear>1970) && releaseYear<= year)
    this.releaseYear = releaseYear;]
```
- 1. Album object should not be null
→ `shouldConstructAlbum`
- 2. Record number should return proper value while adding and updating
→ `recordNumberShouldReturnProperValueAddingAndUpdating`
- 3. Record Number can only accept Alphanumeric, and should not accept special characters
→ `recordNumberCanOnlyAcceptAlphanumericWithSpaceORWithForwardSlash`
- 4. Record number should only accept predefined prefixes ie. ECM, Carmo, RJAL, YAN, Watt, and XtraWatt, otherwise throw illegal argument exception
→ `recordNumberShouldOnlyAcceptPredefinedPrefixWithSpace`
- 5. Record Number prefix is case sensitive
→ `shouldThrowIllegalArgumentOutOfRangeExceptionPrefixCaseIsNotFollowed`
- 6. Record Number can only accept suffix of number, so Illegal Argument Exception is thrown
→ `recordNumberShouldOnlyAcceptSuffixOfNumber`
- 7. Record Number can only accept suffix of a number which might contain forward-slash like "ECM 1064/65"
→ `shouldAcceptProperRecordNumber`
- 8. Should throw Illegal Argument Exception when record number does not have space after the prefixes
→ `exceptionForRecordNoDoesNotHaveSpace`
- 9. Should throw Null Pointer Exception when featured musician is set to null
→ `shouldThrowExceptionWhenSetFeaturedMusicianSetToNull`
- 10. Same name for two musicians should refer same Musician object
→ `twoMusicianNamesShouldReferSameMusician`
- 11. Two Musician Instrument should refer to same musician and musician instrument
→ `twoMusicalInstrumentShouldReferSameMusicianAnd-`
→ `SameMusicalInstrumentOfMusicianInstrumentAttribute`
- 12. Should throw Null Pointer Exception when instruments is set to null
→ `shouldThrowExceptionWhenSetInstrumentsSetToNull`
- 13. Should return proper albumURL when set from ECM website.
→ `shouldReturnProperECMURL`
- 14. Should Throw Unknown Host Exception when invalid URL is set
→ `shouldThrowUnknownHostExceptionWhenInvalidURLIsSet`
- 15. should throw illegal argument exception when hostname does not contain "ecmrecords"
→ `shouldThrowIllegalArgumentOutOfRangeExceptionWhenHostnameDoesNotContainEcmrecords`
- 16. Exception should be thrown if less than 1 track is set
→ `shouldThrowIllegalArgumentOutOfRangeExceptionWhenLessThanOneTrackIsSet`
- 17. Input should be a 4-digit number between 1970 and current year
→ `releaseYearShouldBeBetween1970AndCurrent`
- 18. to check if the set value is correctly returned
→ `shouldReturnCorrectValueWhichIsSetForReleaseYear`
- 19. check album name format
→ `checkAlbumName`
- 20. Should accept proper album name
→ `shouldAcceptProperAlbumName`
- 21. Should reject improper album name with one or multiple invalid letters
→ `shouldThrowIllegalArgumentOutOfRangeExceptionWhenAlbumNameIsSetAnyInvalidLetter`

22. Album Release format should have predefined values
➔ `shouldAcceptValidReleaseFormats`
23. Album Release format should accept predefined values
➔ `shouldRejectInvalidReleaseFormats`
24. Should accept proper genre name or styles (can contain &)
➔ `shouldAcceptProperGenre`
25. Should reject improper Genre
➔ `shouldThrowIllegalArgumentOutOfRangeExceptionWhenGenreSetInvalidValues`
26. Should throw null pointer exception when album genre is set to null
➔ `shouldThrowNullPointerExceptionWhenGenreSetToNull`
27. Should throw null pointer exception when albumReview is set to null
➔ `shouldThrowNullPointerExceptionWhenAlbumReviewSetNull`
28. Same website URL and rating for the album review should refer same Review object
➔ `twoReviewWithSameReviewWebsiteURLandRatingShouldReturnSameReview`
29. Same track name and length should refer to the same Track object
➔ `twoTracksWithSameNameAndLengthShouldReferSameTrack`
30. Should throw null pointer exception when tracks is set to null
➔ `shouldThrowNullPointerExceptionWhenTrackSetToNull`

MUSICIANTEST CLASS

Unit Tests and their respective methods:

1. Musician Name cannot be null
➔ `musicianNameCannotBeNull`
2. Musician name cannot be empty or blank
➔ `MusicianNameCannotBeEmptyOrBlank`
3. should return a musician name
➔ `shouldGetMusicianName`
4. If wrong musician URL is entered which is not responsive an exception should be thrown
➔ `shouldThrowUnknownHostExceptionWhenInvalidURLIsSet`
5. Should accept proper Musician name
➔ `shouldAcceptProperMusicianName`
6. Should reject improper Musician name with multiple invalid letters
➔ `shouldThrowIllegalArgumentOutOfRangeExceptionWhenMusicianNameIsSetALetter`

MUSICIAN INSTRUMENT CLASS

Unit Tests and their respective methods:

1. musician object cannot be null
➔ `musicianCannotBeNull`
2. musical Instrument object cannot be null
➔ `musicalInstrumentCannotBeNull`
3. Same Musician And Musical Instrument Means Same Musician Instrument
➔ `sameMusicianAndMusicalInstrumentMeansSameMusicianInstrument`
4. should return a musician as an object
➔ `shouldGetMusician`
5. should return a musical Instrument as an object
➔ `shouldGetMusicalInstrument`

MUSICAL INSTRUMENT TESTS

Unit Tests and their respective methods:

1. setMusician Instrument Name with null argument should throw null pointer exception
➔ `shouldThrowNullPointerExceptionWhenMusicalInstrumentNameSetNull`
2. Should accept proper Musical instrument name
➔ `shouldAcceptProperMusicalInstrumentName`
3. Should reject improper Musician name with multiple invalid letters
➔ `shouldThrowIllegalArgumentOutOfRangeExceptionWhenTrackNameIsSetAnyInvalidLetter`

Extra Credit:

The unit tests below are for the extra credentials

TRACK UNIT TESTS

Unit Tests and their respective methods:

1. Track object should not be null
→ `shouldConstructTrack`
2. Track name cannot be empty or blank
→ `trackNameCannotBeEmptyOrBlank`
3. `setTrackName` with null argument should throw `NullPointerException`
→ `trackNameCannotBeEmptyOrBlank`
4. Should reject improper track name
→ `shouldThrowIllegalArgumentExceptionWhenTrackNameIsSetAnyInvalidLetter`
5. Should accept proper track name, proper track name might include "" and "-"
→ `shouldAcceptProperTrackName`
6. Should accept proper track length in minute and second format
→ `shouldAcceptProperTrackLengthFormatOfTime`
7. Should throw `IllegalArgumentException` when track length format is not "MM:SS" (assumption 99:59 is max track length)
→ `shouldThrowIllegalArgumentExceptionWhenFormatIsNotMatched`
8. Should throw `IllegalArgumentException` when track length second is set more than 59
→ `shouldThrowIllegalArgumentExceptionWhenLengthSecondIsInvalid`
9. Should accept proper length of tracks
→ `shouldAcceptProperLengthOfTracks`

REVIEW UNIT TEST

Unit Tests and their respective methods:

1. Review object should not be null
→ `shouldConstructReview`
2. Should return proper website url when set
→ `shouldThrowExceptionWhenWebsiteURLSetToNull`
3. Should throw exception when website URL is set to null
→ `shouldThrowExceptionWhenWebsiteURLSetToNull`
4. Should Throw Unknown Host Exception when invalid websiteURL is set
→ `shouldThrowUnknownHostExceptionWhenInvalidWebsiteURLIsSet`
5. Should throw null pointer exception when rating is set to null
→ `shouldThrowIllegalArgumentExceptionWhenRatingIsSetToNegative`
6. should throw illegal argument exception when rating is set to less than 0.00 or more than 5.00
→ `shouldThrowIllegalArgumentExceptionWhenRatingValueSetOutOfRange`
7. Should accept parameter of integers when required
→ `shouldAcceptIntegerOfWholeNumber`
8. Review cannot be empty or blank
→ `reviewCannotBeEmptyOrNull`
9. Same ratingURL and same rating means same Review
→ `sameRatingURLandSameRatingMeansSameReview`

Contribution Declaration Form

(to be completed by all team members)

Please fill in the form with the contribution from each student towards the assignment.

1 NAME AND CONTRIBUTION DETAILS

| Student ID | Student Name | Contribution Percentage |
|------------|---------------------|-------------------------|
| 29994985 | Neha Jere | 23% |
| 29600189 | Zhibin Kang | 23% |
| 29973740 | Aishwarya Karkera | 23% |
| 30270111 | Farhad Ullah Rezwan | 31% |

Contributions:

Farhad Ulla Rezwan:

- Album class
- Extra Credit points
- Musical Instrument test class
- Report

Neha Jere:

- Album class
- Musical Instrument Class
- Report

Aishwarya Karkera:

- Musician test class
- Report

Zhibin Kang:

- MusicianInstrument test class
- Report

2 DECLARATION

We declare that:

- The information we have supplied in or with this form is complete and correct.
- We understand that the information we have provided in this form will be used for individual assessment of the assignment.

3 SIGNATURE

Signatures

Neha Jere

Aishwarya Karkera

Farhad Ullah Rezwan

Zhibin Kang

DATE :

19 /04 /2020 2019