



# Code Sniffer

Team-202 members :

Hao Wu

Anubhuti Vyas

Sidharth Thapar

Fibin Francis Assissi

# System Functionalities - Dashboard

Hi Anubhuti! How are you doing today?

Just select the semester, and there you go!

Select course:  
**Spring2016**  
semester

Select course:  
**Summer2017**  
semester

Select course:  
**Fall2016**  
semester

Select course:  
**Spring2018**  
semester

Select course:  
**Summer2016**  
semester

Select course:  
**Fall2017**  
semester

# System Functionalities - Student Dashboard

Hi ! Here are your courses for this semester

Algo

- Algorithm-Assgn1
- Algo-Assgn2
- hw1

Submit

Submit

Submit

MR

IR

Add Course

Choose upload method for :

☒ Zip File ☐ GitHub Link

Choose file

Choose File No file chosen

Upload File

# System Functionalities - Student Dashboard

Hi ! Here are your courses for this semester

MR

Algo

- Algorithm-Assgn1
- Algo-Assgn2
- hw1

Submit

Submit

Submit

Add Course

Choose upload method for :

☐ Zip File ☒ GitHub Link

Enter GitHub URL

GitHub URL

Upload File

# System Functionalities - Faculty Dashboard

Hi ! Here are your courses!

MR	Add Course
ML	
MSD	
Algo	

- Algorithm-Assgn1 ✓ 78
- Algo-Assgn2 ✓ 45
- hw1 ✓ 50

Add New Assignment

Create Course	
Course Name	<input type="text" value="Course name"/>
Send report to	<input type="text" value="e-mail of recipient"/>
Create	

# System Functionalities - Faculty Dashboard

## Code Sniffer

### Report for Hao and Anubhuti

Matches for Algo-Assgn2-Hao &  
Algo-Assgn2-Anubhuti

48.2%

Algo-Assgn2-Hao (37.80488%)	Algo-Assgn2-Anubhuti (66.666664%)	Tokens
<a href="#">hw5\NonEmptyRoster.java(16-38)</a>	<a href="#">hw4\EmptyRoster.java(13-35)</a>	13
<a href="#">hw5\NonEmptyRoster.java(128-145)</a>	<a href="#">hw4\EmptyRoster.java(91-108)</a>	11
<a href="#">hw5\NonEmptyRoster.java(152-191)</a>	<a href="#">hw4\EmptyRoster.java(109-147)</a>	38

#### hw5\NonEmptyRoster.java

```
import java.util.*;
import java.util.Iterator;
import java.util.List;

//Constructor template for NonEmptyRoster:
//new NonEmptyRoster (Set<Player> p)
//Interpretation:
//p represents the set of Players in this Roster

public class NonEmptyRoster implements Roster{
    private final Set<Player> players;

    public NonEmptyRoster(Set<Player> p) {
        this.players = p;
    }

    // Returns a roster consisting of the given player together
    // with all players on this roster.
    // Example:
    // r.with(p).with(p) => r.with(p)

    public Roster with (Player new_player) {
        Set<Player> new_players = getPlayers(this.players);
```

#### hw4\EmptyRoster.java

```
import java.util.*;

//Constructor template for EmptyRoster:
//new EmptyRoster ()

public class EmptyRoster implements Roster{
    private final Set<Player> players;

    public EmptyRoster() {
        this.players = Collections.emptySet();
    }

    // Returns a roster consisting of the given player together
    // with all players on this roster.
    // Example:
    // r.with(p).with(p) => r.with(p)

    public Roster with (Player new_player) {
        Set<Player> new_players = getPlayers(this.players);
        new_players.add(new_player);
        Roster r = new NonEmptyRoster(new_players);
```

# Email Notification



From: <[sid.codesniffer@gmail.com](mailto:sid.codesniffer@gmail.com)>

Date: 18 April 2018 at 17:39

Subject: Plag detected in Algorithm-Assgn1 between Anubhuti and Hao

To: [anubhuti.vyas.28@gmail.com](mailto:anubhuti.vyas.28@gmail.com)

Codesniffer found plagiarized submission with similarity score 100.0%. Click the below link to view the full report

[https://s3.amazonaws.com/codesniffer-reports/user2\\_sem6\\_course4\\_asgmtAlgorithm-Assgn1\\_sub27\\_user3\\_sem5\\_course4\\_asgmtAlgorithm-Assgn1\\_sub26/match0.html](https://s3.amazonaws.com/codesniffer-reports/user2_sem6_course4_asgmtAlgorithm-Assgn1_sub27_user3_sem5_course4_asgmtAlgorithm-Assgn1_sub26/match0.html)

# Functionalities Achieved



Basic Expectations (100%)	Stretch Goals Achieved
CRUD for semester, course, assignments	Send Notification Email To Faculties
Incrementally compare all submissions	Compare 2 languages
Multiple submission for students	Github upload
Side-by-side code comparison	Compare across previous semesters





# App Quality

## UI Quality:

1. The app has an intuitive UI.
2. Suggestions from TA and professor throughout the sprints were incorporated to improve the usability of the tool.



# App Quality

## Code Quality:

1. MVC Architecture
2. Modularity. Frontend and backend are isolated modules.
3. Frontend as well as the backend code was validated.
4. Http Session to implement Session Management



# App Quality

## Code Quality:

1. Javadocs were used to make the code more readable.
2. The code is well tested using Mockito.
3. SonarQube quality gate ensured all quality metrics are achieved before the code is pushed to the git, at all times.
4. There was a significant improvement in the quality of the code for the final version.

## Quality Gate Passed

## Bugs &amp; Vulnerabilities

0 A

Bugs

0 A

Vulnerabilities

Leak Period: since previous version  
started a month ago

0 A

New Bugs

0 A

New Vulnerabilities

## Code Smells

6h A

Debt

started a month ago

31

Code Smells

6h A

New Debt

31

New Code Smells

## Coverage



95.7%

Coverage

156

Unit Tests

96.0%

Coverage on  
482 New Lines to Cover

## Duplications



1.6%

Duplications

5

Duplicated Blocks

0.8%

Duplications on  
5.4k New Lines

## Plagiarism detector

S 1.6k

Lines of Code

Java 1.5k

XML 122

No tags

## Activity



April 18, 2018

0.0.1-SNAPSHOT

April 18, 2018

Project Analyzed

April 18, 2018

Project Analyzed

[Show More](#)

Quality Gate

(Default) SonarQube way

Quality Profiles

(Java) Sonar way

(XML) Sonar way

[Home](#)[Sources](#)

Key


com.msdp.project.codesniffer

# Process and Team work





- AGILE methodology with sprints extending 2 weeks
- Daily standups through slack
- JIRA updated during sprint planning
- Shortcoming - Initial sprints lacked proper planning, due to which there were unequal distribution of work but overcame right after feedback from sprint1
- Solution -In addition to actual sprint review, we had mini-reviews at regular intervals that ensured equal contribution


# Automation


 **Jenkins** 2  [Fibin Francis Assissi](#) | [log out](#)


Jenkins ▾ MSD project ▾ master ▾ [ENABLE AUTO REFRESH](#)


 Up


 **Status**


 Changes


 Build Now


 View Configuration

 Full Stage View

 SonarQube


 Open Blue Ocean

 GitHub

 Pipeline Syntax

## Branch master

Full project name: MSD project/master

 [Recent Changes](#)

## Stage View

Average stage times:  
(Average **full** run time: ~2min 53s)

	Declarative: Checkout SCM	Declarative: Agent Setup	Build	Test	SonarQube	Quality
#66 Apr 18 15:36 1 commits	453ms	491ms	45s	16s	43s	32s
#65 Apr 18 15:26 1 commits	544ms	357ms	1min 14s	32s	1min 14s	32s
#64 Apr 18 14:44 1 commits	2s	1s	46s	16s	40s	32s

### Build History

[trend ▾](#)

find

#66 Apr 18, 2018 8:36 PM

#65 Apr 18, 2018 8:26 PM

#64 Apr 18, 2018 7:44 PM

#63 Apr 18, 2018 5:25 PM

# Technology Transfer



- Actively Hosted on Amazon AWS
  - Ready to use by clients
  - URL in the description
- Next Steps:
  - OAuth login function
  - Compare different sections of same course
  - Users testing for UI improvements
- Easy to be extended by the successors:
  - Extensible modular design
  - MVC Design Pattern
  - Formal Documentation, Javadoc