Code Sniffer

Team-202 members :

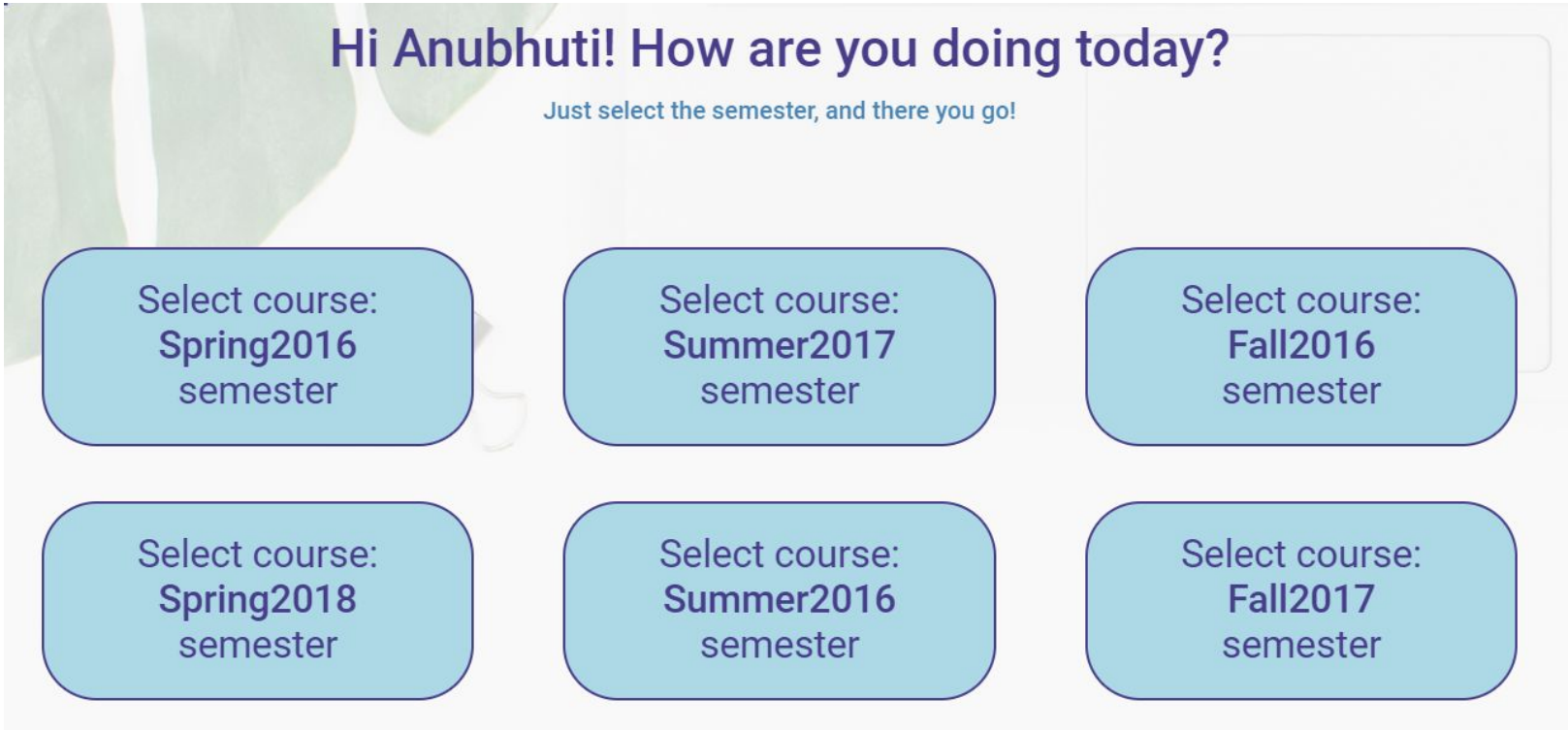
Hao Wu

Anubhuti Vyas

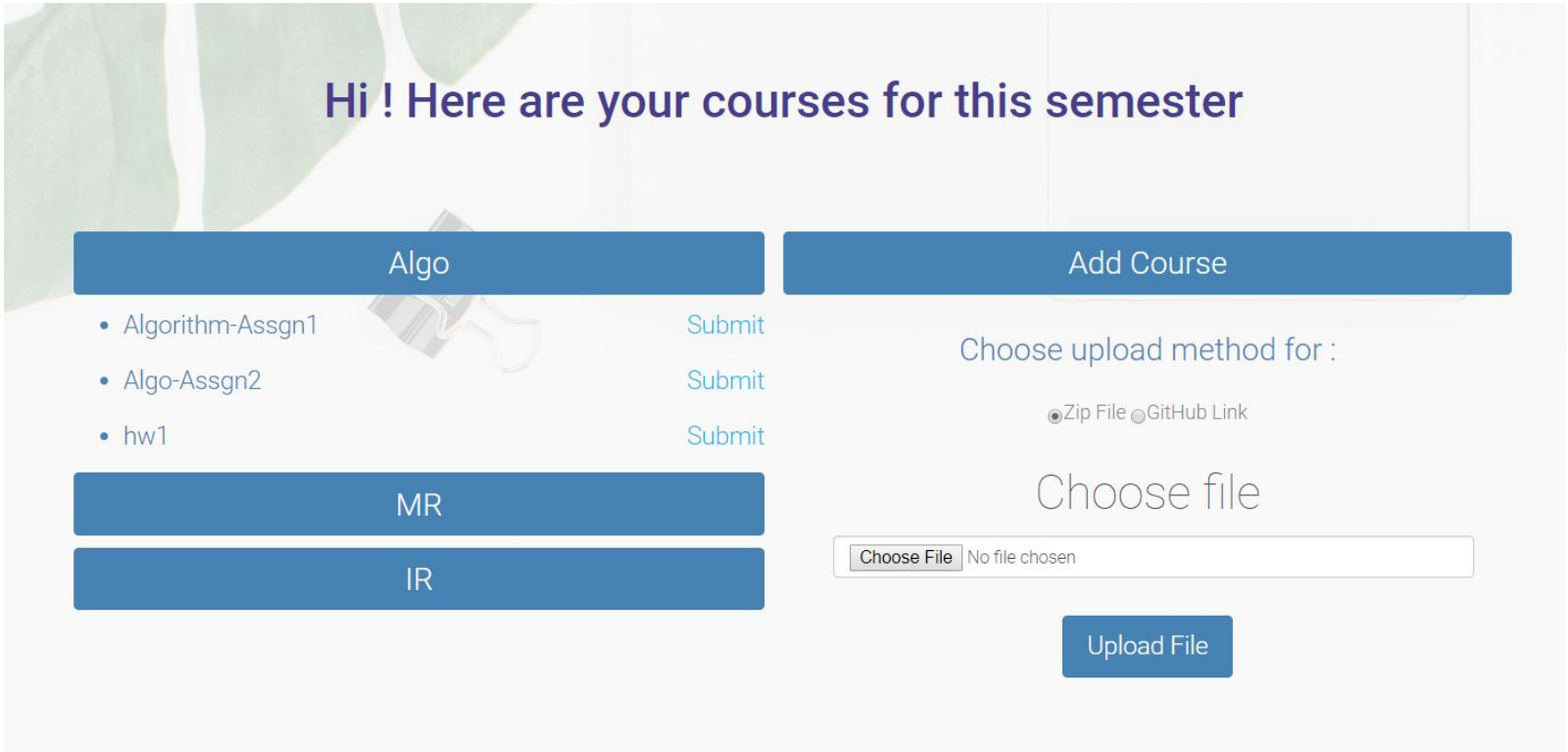
Sidharth Thapar

Fibin Francis Assissi

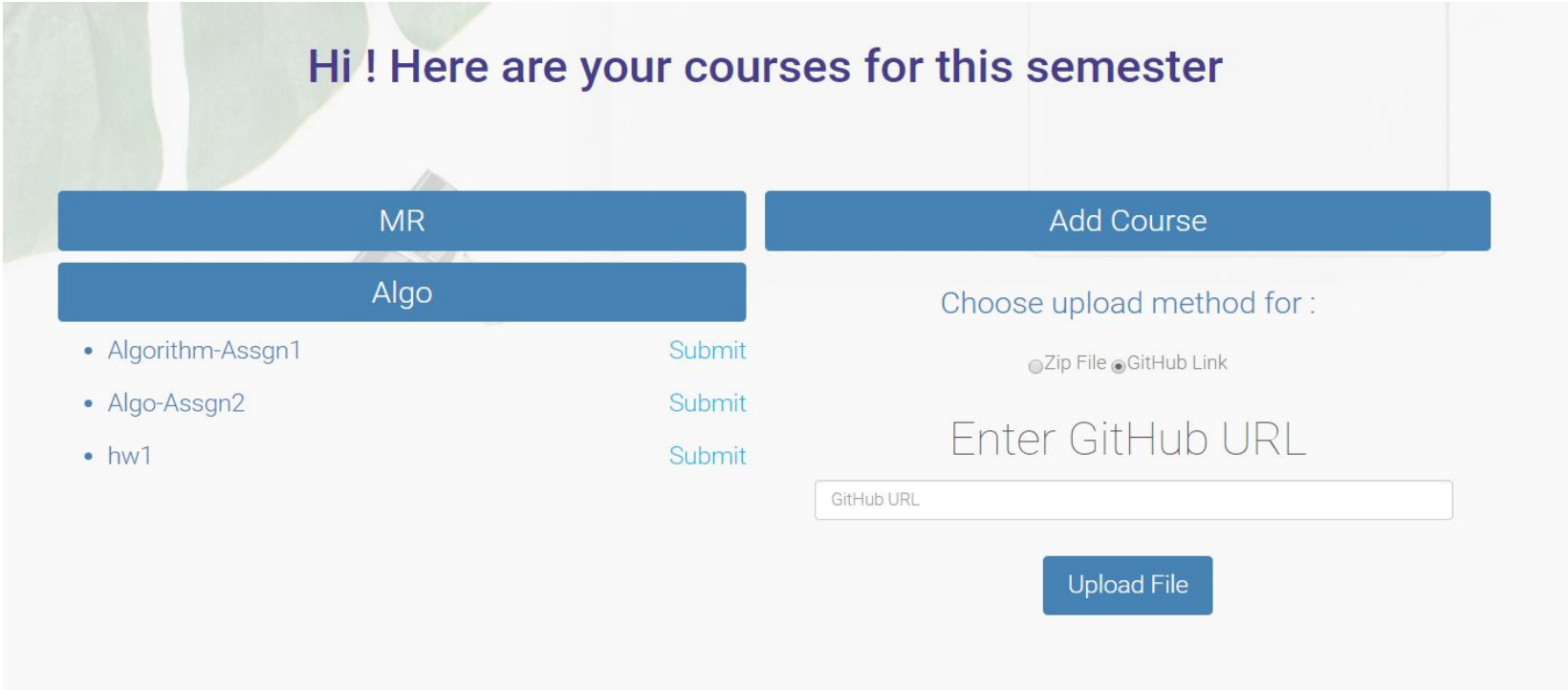
System Functionalities - Dashboard



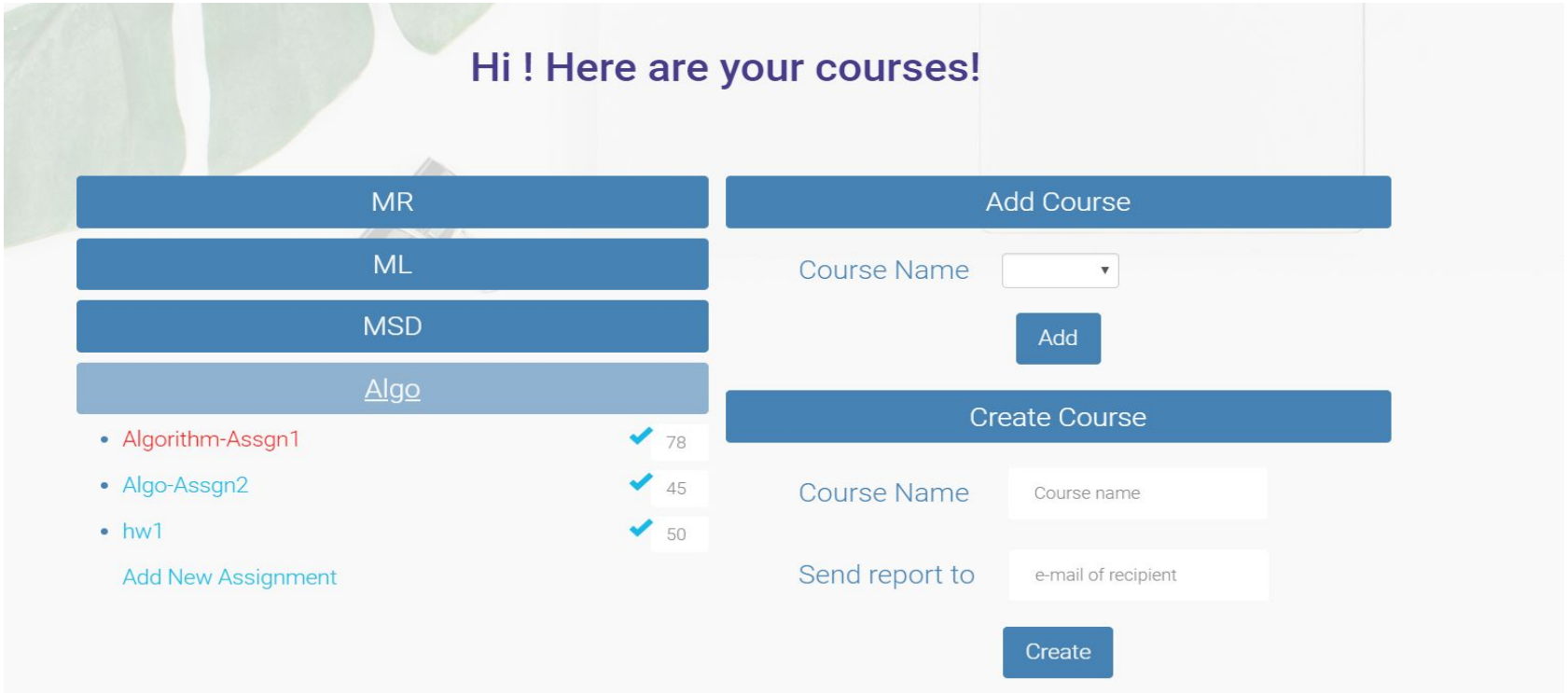
System Functionalities - Student Dashboard



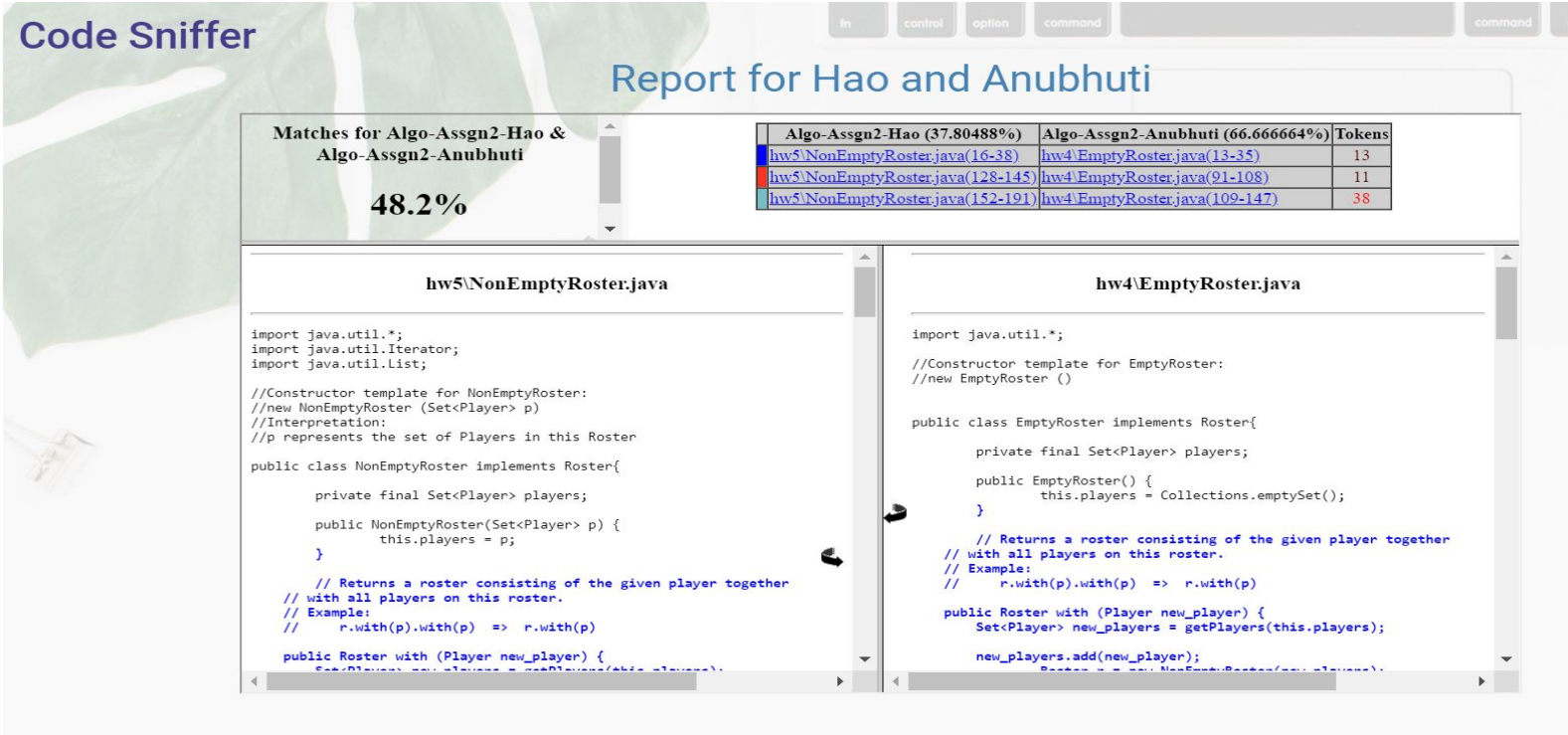
System Functionalities - Student Dashboard



System Functionalities - Faculty Dashboard



System Functionalities - Faculty Dashboard



Email Notification

From: <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

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

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

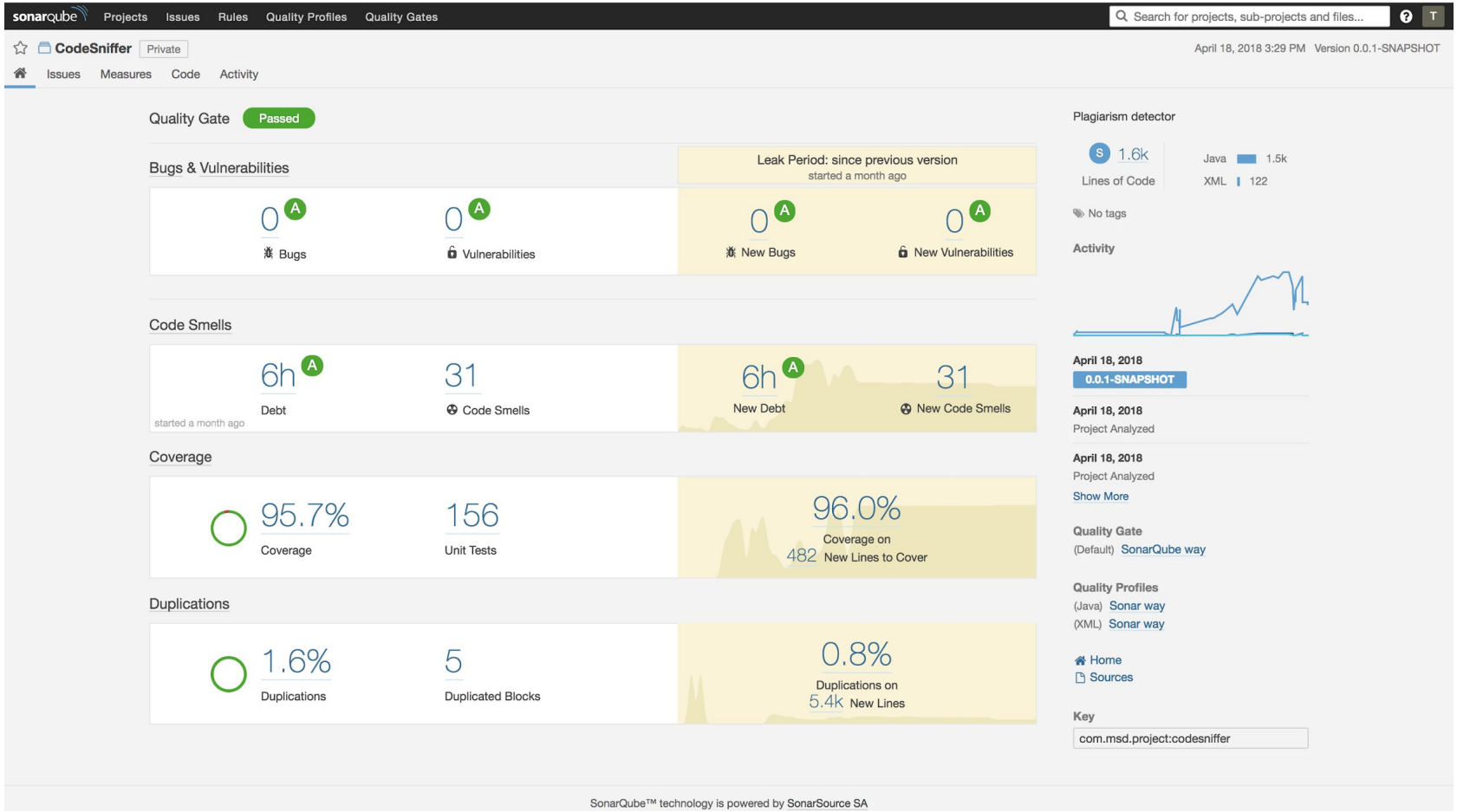
Code Quality:

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:

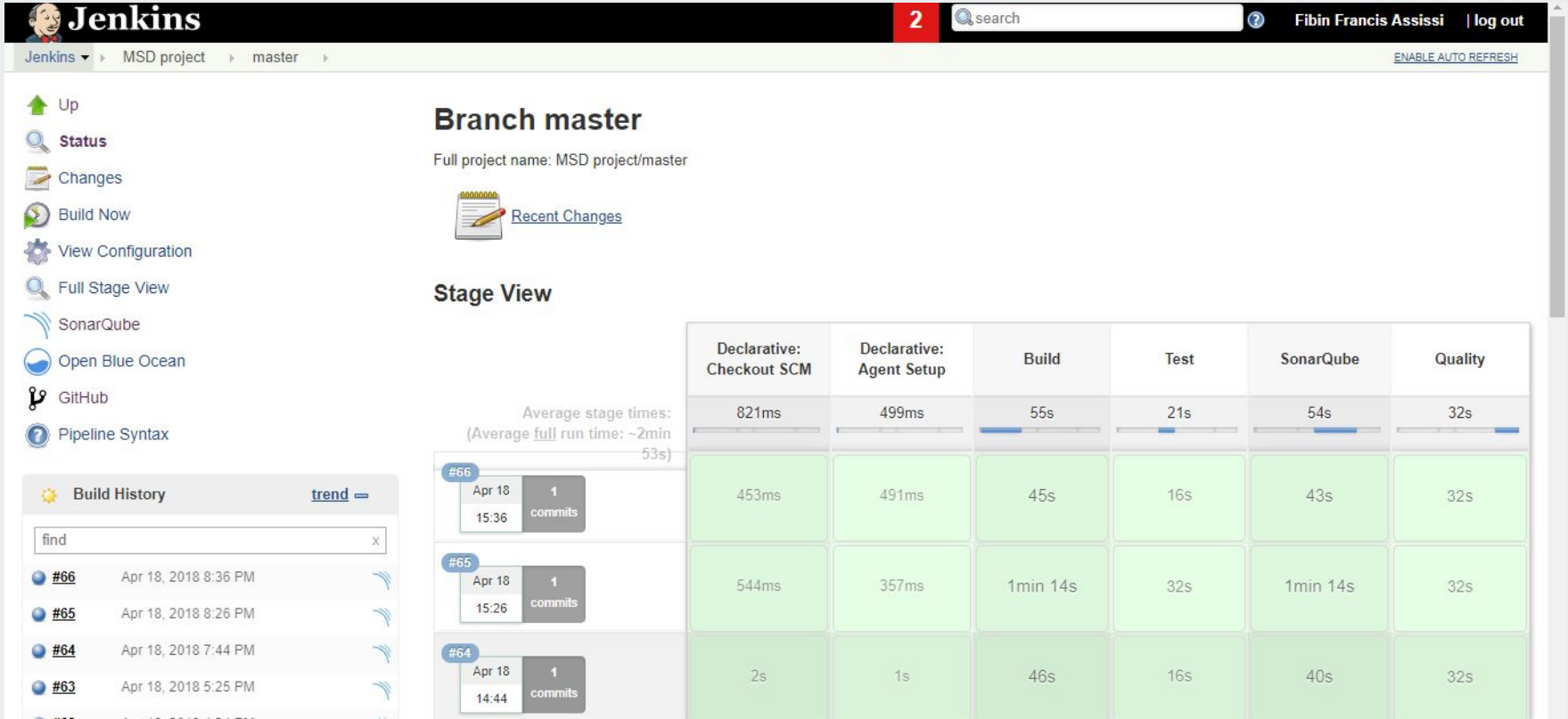
1. Javadocs were used to make the code more readable.
2. The code is well tested using Mockito.
3. SonarQube quality gate ensured all quality metrics are achieved before the code is pushed to the git, at all times.
4. There was a significant improvement in the quality of the code for the final version.


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Search for projects, sub-projects and files...
T
CodeSniffer
Private
April 18, 2018 3:29 PM Version 0.0.1-SNAPSHOT
Issues
Measures
Code
Activity
Quality Gate
Passed
Bugs & Vulnerabilities
Leak Period: since previous version
started a month ago
0 A
Bugs
0 A
Vulnerabilities
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.msd.project:codesniffer
SonarQube™ technology is powered by SonarSource SA

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
search
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
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
Branch master
Full project name: MSD project/master
Recent Changes
Stage View
Declarative: Checkout SCM
Declarative: Agent Setup
Build
Test
SonarQube
Quality
Average stage times:
(Average full run time: ~2min 53s)
821ms
499ms
55s
21s
54s
32s
#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

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