



Final Presentation of Plagiarism Checker

CS5500 Team 106

System Functionality



Register
Login
Logout

Why it is needed for customers?

Because customers want to see plagiarism checking history, and historical report is identified by its ID and the user who creates it.

Aka, keep tracking.

Register/Login/Logout

CS106-9: Log in functionality - front end

CS106-10: Log in functionality - back end

CS106-21: React unable to call end point on AWS

CS106-34: Front end refreshing automatically

CS106-36: Fix the confirm password

CS106-42: Securing login sessions

CS106-59: Login test case impacted

CS106-70: Login uid bug

[Log In](#) [Register](#)

User Registration

Email *

Name *

Password *

Confirm *

Please fill in the form

Welcome manpreet [Log out](#)

[Log In](#) [Register](#)

User Login

Email *

Password *

Please fill in the form

Features

- Email validation: not null, must be in the format like user@example.com
- Name validation: not null
- Password length validation
- Password confirmation validation
- User cannot register or login if the information is not complete

[Log In](#) [Register](#)

User Registration

Email *

Name *

Password *

Confirm *

Please fill in the form

Welcome manpreet [Log out](#)

[Log In](#) [Register](#)

User Login

Email *

Password *

Please fill in the form



Start New Run

Plagiarism Checking → Run

Sample testing Git Repos:

<https://github.com/bharat94/testRepo1.git>

<https://github.com/bharat94/testRepo2.git>

The main and the most important functionality of the application.

Customers must start a new plagiarism checking in this user interface.

ymoinres

Start New Run

CS106-41: Plagiarism run framework

CS106-58: Plagiarism run service

CS106-71: Fix UI

Add New Run

Step 1: Add Student (GitHub Repo)

Add Student

Example: <https://github.com/user/exampleRepo.git>

Student 1 GitHub Link

Student 1 Name

Step 2: Define Run

Advanced

Description: Run Description

Step 3: Adjust Strategies Weight

Reset

Levenshtein

33%

LCS

33%

Cosine

33%

Add run ID and a description

Cancel

Features

- Run service will be disabled if there are less than two students
- Must finish adding students then add description
- Users are able to adjust weight of strategies
- Information must be complete to start the run
- GitHub repo URL validation

Add New Run

Step 1: Add Student (GitHub Repo)

Add Student

Example: <https://github.com/user/exampleRepo.git>

Student 1 GitHub Link

Student 1 Name

Step 2: Define Run

Advanced

Description: Run Description

Step 3: Adjust Strategies Weight

Reset

Levenshtein

33%

LCS

33%

Cosine

33%

Add run ID and a description

Cancel



View Reports

Another main and important functionality of the application.

Customers can get a report of the run they created in this user interface.



View Reports

CS106-32: Implement professors dashboard

CS106-46: A simple dashboard

CS106-47: Reports services

CS106-48: An interactive dashboard

CS106-52: Pull Student submissions from git

CS106-41, 51, 53: Three complicated strategies

CS106-64: Call plagiarism and report backend endpoints

Statistics of 17

Run Description: Sample Description

Student1	Student2	File1	File2	Percentage	Severity	GitDiff
Student1	Student2	student1-file1.py	student2-file2.py	0.16	Low	View ▾
Student1	Student2	student1-file1.py	student2-file1.py	0.16	Low	View ▾
Student1	Student2	student1-file2.py	student2-file2.py	1	High	View ▾
Student1	Student2	student1-file2.py	student2-file1.py	0.81	High	View ▾



Admin Dashboard

Sample Admin credentials:

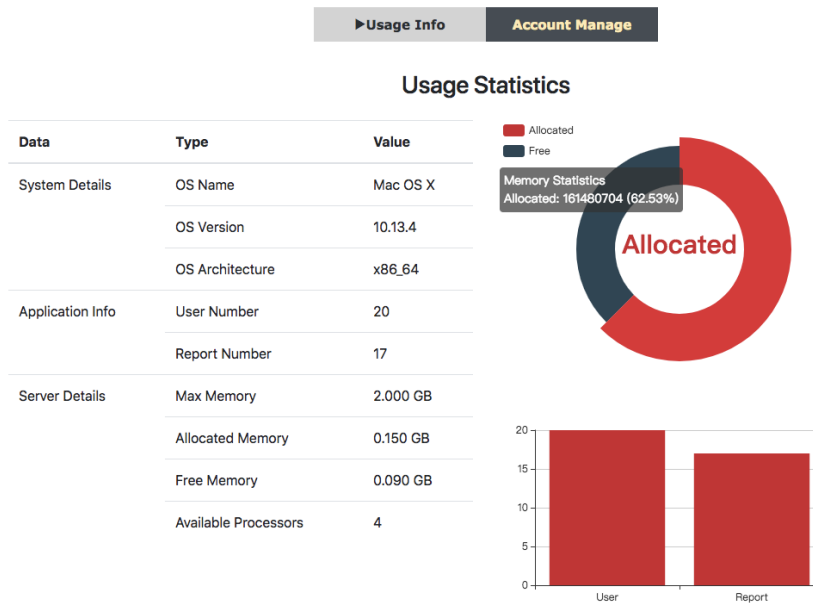
admin2@example.com
123456

Why customers need this?

An application must involve an administrator who is responsible for getting usage statistics and managing users (blocking users).

Admin Dashboard

CS106-33: Implement admin's dashboard





Managing/Blocking Users

CS106-106

Usage Info

► Account Manage

Add Admin User

Email

user@example.com

*

Password

6 to 15 characters

*

Confirm

Confirm Password

*

Fill in info

Remove user

Email

user@example.com

*

Search

Job Quality



Defects Fixing

Sprint 1

CS106-21 *	React unable to call end point on aws	🔴 Bug
CS106-34 *	Front end refreshing automatically	🔴 Bug
CS106-36 *	Fix the confirm password	🔴 Bug

Sprint 2

CS106-59 *	Login test case impacted	🔴 Bug
CS106-62 *	List entity error	🔴 Bug
CS106-70 *	login uid bug	🔴 Bug
CS106-71 *	Fix UI	🔴 Bug

Sprint 3

CS106-82 *	Fix to the weighted polynomial function.	🔴 Bug
CS106-106 *	Add restrictions to "add new run"	🔴 Bug



Defects Fixing

❑ ⬆ [CS106-158](#) No validation check to accept student repositories (Accepts fake repositories)

❑ ⬆ [CS106-161](#) Need to check whether the report object is null in js files.

❑ ⬆ [CS106-163](#) Exact same github url for 2 students, gives 85% plagiarism

❑ ⬇ [CS106-171](#) Empty test in Comparison Test class

❑ ⬆ [CS106-184](#) Unusual behavior of a system (log out issue)

❑ ⬇ [CS106-186](#) Add run functionality accepts any kind of repository -not only python

❑ ⬆ [CS106-204](#) Code review problems in testing classes

❑ ⬆ [CS106-206](#) Test case out unpredicted

❑ ⬆ [CS106-207](#) Maybe an issue in comparison algorithm



Feed back from TA

Sprint 2 Group Grade Team 106:

Functionality Expectations Achieved:

Comments on Functionality Expectations:

UI Comments:

1. The link ""View UI as non-wireframes"" does not work
2. Admin login does not work
3. UI needs prompts; labels and user messaging
4. Comparison UI (Add Run) has no labels or instructions for inputs.

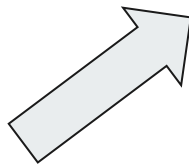
Sprint 3 Group Grade Team 106:

Functionality Expectations Achieved:

Sets of Test Projects (out of 6): 6

Comments on sets of Test Projects: 6 sets provided ar

Comments on Functionality Expectations: "URL: http://
UI is clean and nice since Sprint 2. Need to tweak sc



Process and Teamwork

Work as a Team!

Ananta: GitDiff, two comparison strategies, MOSS training

and AST Generation

Bharat: Back-end, end points, detection engine, dev environment

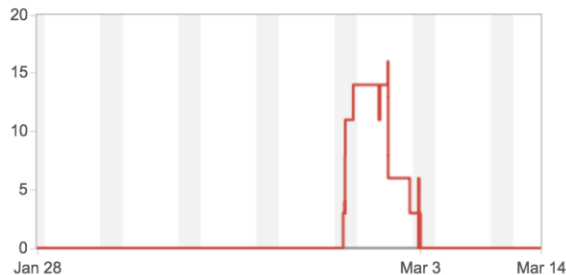
Junhao: whole front-end, one comparison strategy and paperwork

Raghu: Integration, end points and back end, use cases

and session management

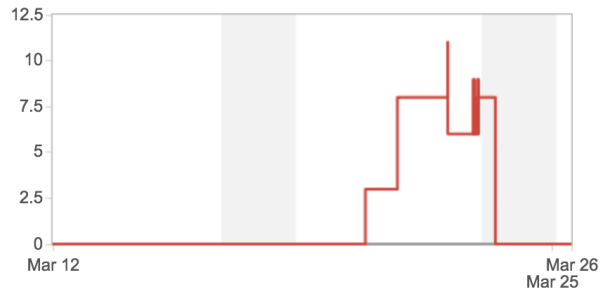
Sprint 1 ▾

Closed Sprint, ended by [Raghavendra Venkatesh](#) 28/Jan/18 8:00 AM - 14/Mar/18 10:56 AM



Sprint 2 ▾

Closed Sprint, ended by [Junhao Qu](#) 12/Mar/18 11:00 AM - 26/Mar/18 9:50 AM



Build & Test Automation



Technology Transfer



Hand Over to Clients

1. The application is well deployed on *AWS* and is able to access on internet. It is in good shape to deliver to customers.
2. We suggest customers to clone the code, deploy Database attributes, use maven to build and generate a jar file on local machine and deploy the jar file.