Sherlock

TEAM 211

HARSH MEHTA, SURAJ NAIR, SHANTANU KAWLEKAR & SANDIP SAMAL

Algorithm

- Zhang and Shasha algorithm
  - Computes the edit distance between the two given trees, as well as a small set of utilities to make its use convenient.
- Moss (for a Measure Of Software Similarity) is an automatic system for determining the similarity of code.

Algorithm

- Linear Regression:
  - Data collection: 40 sample training data.
  - MOSS is used as a standard against our features. Features here include the result of Zhang and Shasha algorithm as well as percentage of lines copied.
  - Zhang and Shasha algorithm is given a polynomial weight of 0.8 and 0.2 weight for percentage of lines copied.

System Functionalities

FEATURES, WORKING AND EVALUATION OF THE SYSTEM

Major Functionalities

- Student
  - CRUD
  - Assignment Submission
- Instructor
  - CRUD
  - Automated Incremental Plagiarism Generation
  - Email with Report link
- Admin
  - User CRUD
  - Setting the semester

Evaluation

- Did the team accomplish a sufficient amount of functionality?
  - Support for Multiple Languages
  - OAuth Single Sign-on
  - Linear Regression
  - Informative Email
  - Multiple Submissions for Students

Evaluation

- Did the team achieve what it set out to do?
    - From the start, our vision was to develop a real-world application that can be used by an Instructor.
    - Our intent was to create a homework submission portal that supports plagiarism detection.
    - We made sure that the plagiarism detection is accurate and that our UI is user friendly.
    - We set out to create a user friendly plagiarism detection system that can be used in the real-world and we believe that we have been successful in achieving the same.

Evaluation

- Is it useful for the client?
  - Yes, the system is readily available as a web-app for the client to use it in a real world scenario.
  - Since our system integrates homework submission as well as plagiarism detection, it automates the plagiarism detection process end-to-end requiring no manual involvement from the student as well as the instructor.
  - This saves time as well as money for the client.

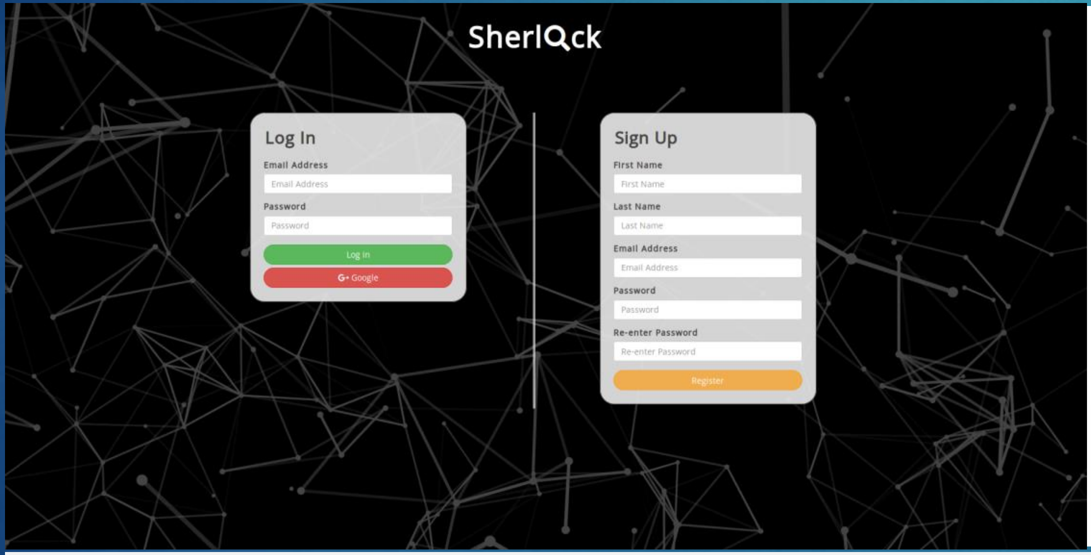
Evaluation

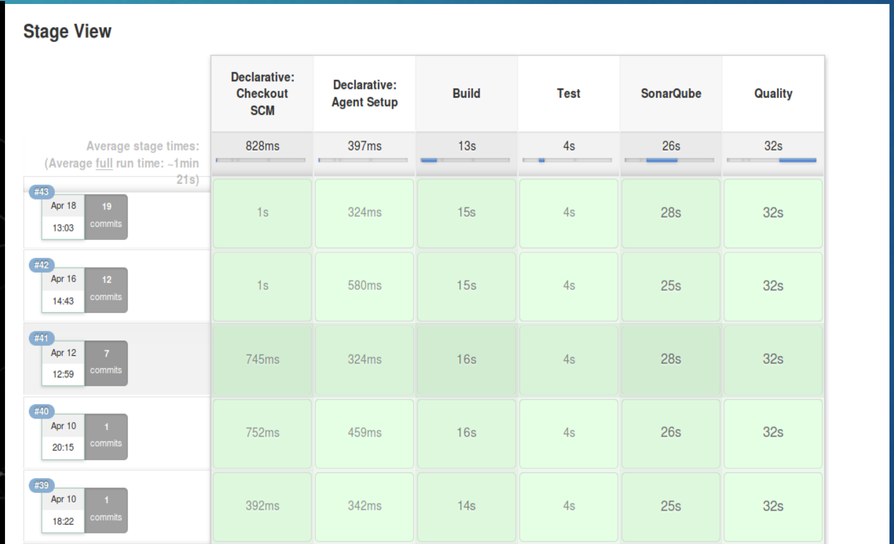
- Is there evidence from the backlog/requirements to support these claims?
  - Over the course of three months, we have created and closed several jira tickets for the development of our application.
  - We have also fixed the bugs reported by the testers during the bug hunt.
  - Furthermore, we have successfully completed the majority of stretch goals provided by the professors.

Job Quality

Work quality and team’s performance.

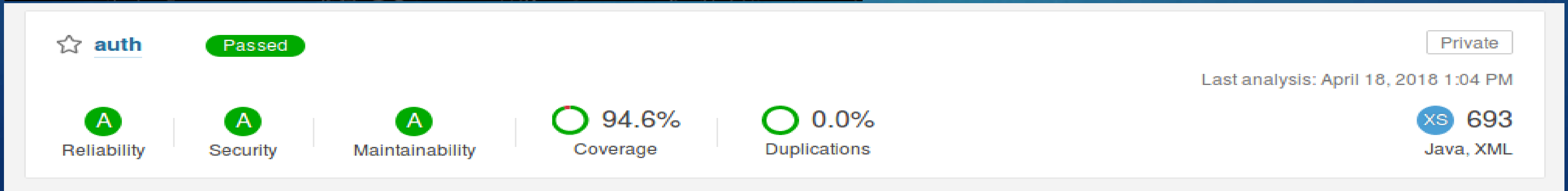
- Did the team present evidence that it did a quality job?
  - We have refined our work with the help of sprint review comments
  - Accomplished stretch goals which adds significantly to the usability of our product.





Stage View

	Declarative: Checkout SCM	Declarative: Agent Setup	Build	Test	SonarQube	Quality
Average stage times: (Average full run time: ~1min 21s)	828ms	397ms	13s	4s	26s	32s
#43 Apr 18 13:03 19 commits	1s	324ms	15s	4s	28s	32s
#42 Apr 16 14:43 12 commits	1s	580ms	15s	4s	25s	32s
#41 Apr 12 12:59 7 commits	745ms	324ms	16s	4s	28s	32s
#40 Apr 10 20:15 1 commits	752ms	459ms	16s	4s	26s	32s
#39 Apr 10 18:22 1 commits	392ms	342ms	14s	4s	25s	32s



auth Passed

Private

Last analysis: April 18, 2018 1:04 PM

A Reliability | A Security | A Maintainability | 94.6% Coverage | 0.0% Duplications | XS 693 Java, XML

- Did the team improve its performance over time?
  - As a team we believe that we had an upward trajectory in terms of performance.
  - Our Motto: Learn as we go
  - Worked better as we got to know each other very well over time.
  - Worked coherently towards achieving our goals

Process And Teamwork

Process usage, organization and team’s combined efforts

- Did the team work as a team?
  - All of us coordinated well with each other and were clear about our roles.
  - We were there to cover up on each other’s behalf in times of unavailability.
  - Pair programming and learning from each other.
  - Working diligently, communicating our availability and asking for help when needed.
  - Rotating our roles

- Did the team use process well or was the project mis/mal organized?
  - Participating into team meetings over hangouts, reporting work and providing inputs/perspective over discussed problems.
  - Creating pull requests and reviewing code before merging it with master branch.
  - Creating issues on Jira and assigning them to team members.
  - Jenkins, Sonarqube for automation and quality check.
  - Jenkins, GitHub integration with Slack

- Was the team able to automate the build, test, and promote processes?
  - We did setup Jenkins for automating our build, test and Sonarqube for code quality inspection.
  - It helped us detect and resolve errors in our code as soon as we wanted to integrate any new feature.
  - This saved us time and efforts required to resolve these issues later if we wouldn’t have automated.
  - Saved time in getting our application up and running with automated deploy stage.

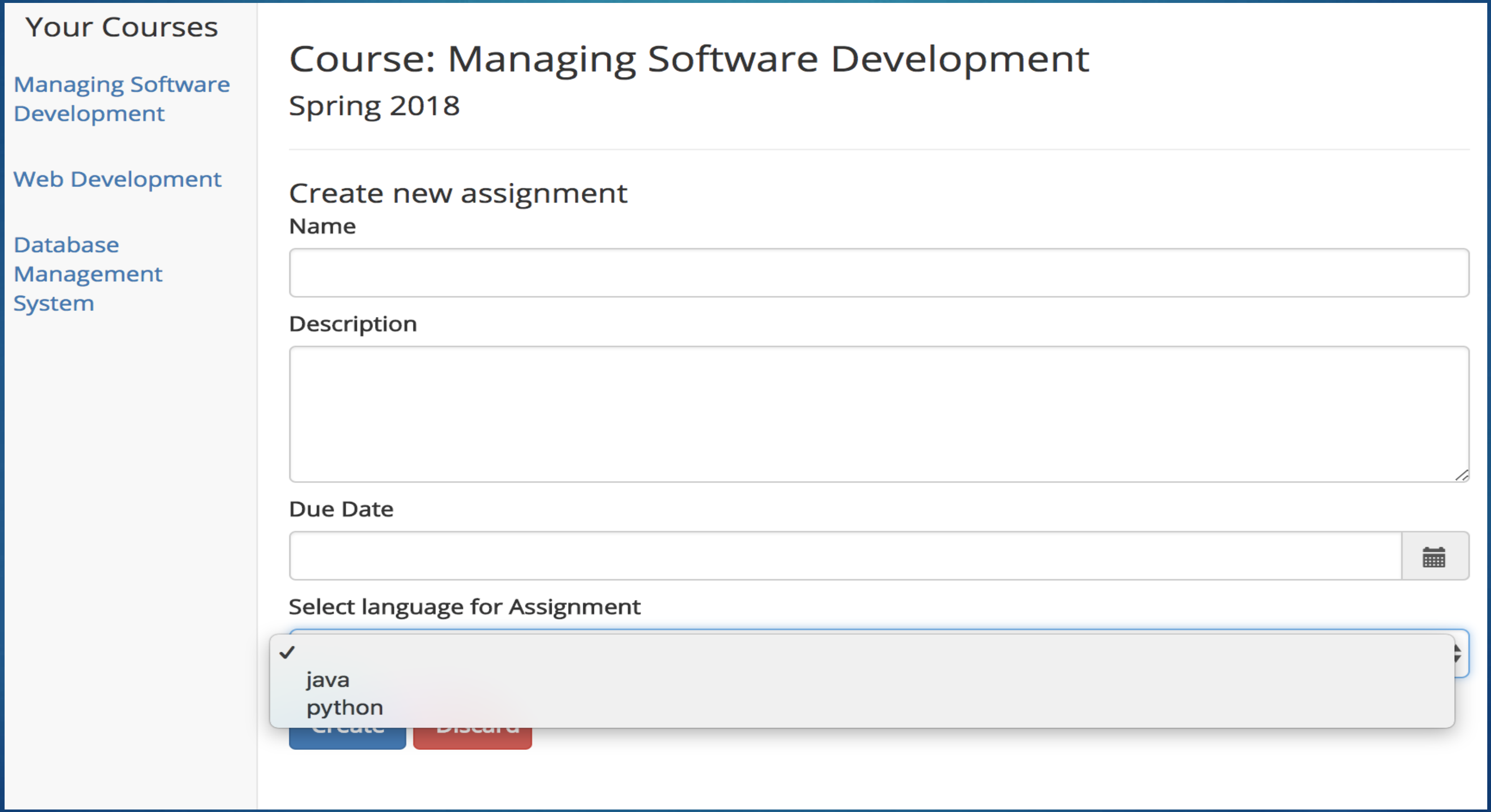
- Did the team recognize short-comings? If so, how well did they work around the challenges?
  - We didn’t have prior experience with some of the technologies that we have used in the development of this project.
  - We responded appropriately to this, we learned and got better at these technologies over time.
  - As a team, we adapted well to the situations.

Technology Transfer

SYSTEM HIGHLIGHTS AND FUTURE SCOPE

Is the system ready for delivery?

Supports more than one language



Your Courses

Managing Software Development

Web Development

Database Management System

Course: Managing Software Development

Spring 2018

Create new assignment

Name

Description

Due Date

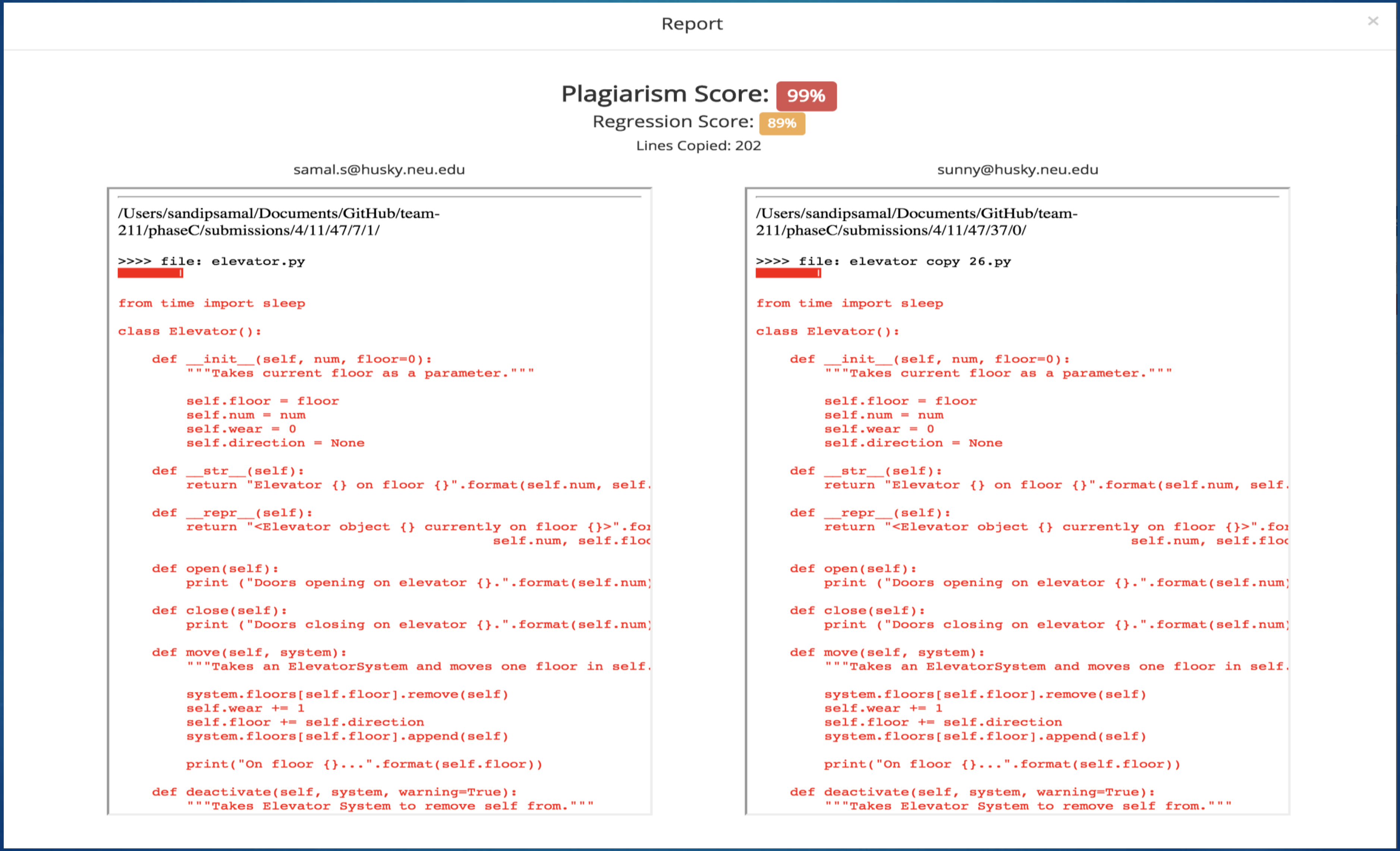
Select language for Assignment

✓

java

python

Detailed plagiarism reports for better judgement



Plagiarism Score: 99%

Regression Score: 89%

Lines Copied: 202

samal.s@husky.neu.edu

/Users/sandipsamal/Documents/GitHub/team-211/phaseC/submissions/4/11/47/7/1/

```
>>>> file: elevator.py

from time import sleep

class Elevator():

    def __init__(self, num, floor=0):
        """Takes current floor as a parameter."""

        self.floor = floor
        self.num = num
        self.wear = 0
        self.direction = None

    def __str__(self):
        return "Elevator {} on floor {}".format(self.num, self.

    def __repr__(self):
        return "<Elevator object {} currently on floor {}>".for
                                    self.num, self.floo

    def open(self):
        print ("Doors opening on elevator {}.".format(self.num)

    def close(self):
        print ("Doors closing on elevator {}.".format(self.num)

    def move(self, system):
        """Takes an ElevatorSystem and moves one floor in self.

        system.floors[self.floor].remove(self)
        self.wear += 1
        self.floor += self.direction
        system.floors[self.floor].append(self)

        print("On floor {}...".format(self.floor))

    def deactivate(self, system, warning=True):
        """Takes Elevator System to remove self from."""
```

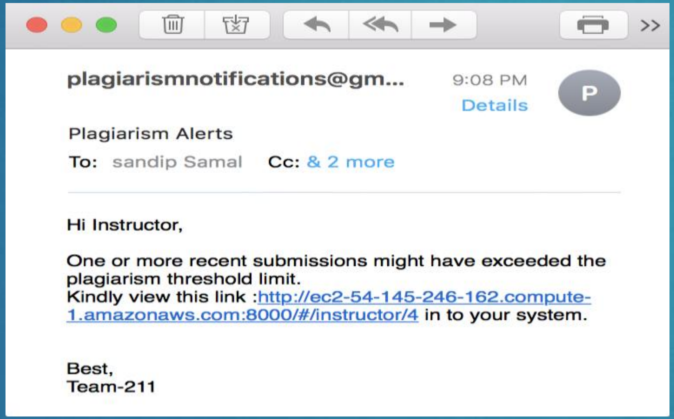
sunny@husky.neu.edu

/Users/sandipsamal/Documents/GitHub/team-211/phaseC/submissions/4/11/47/37/0/

```
>>>> file: elevator copy 26.py

from time import sleep

class Elevator():

    def __init__(self, num, floor=0):
        """Takes current floor as a parameter."""

        self.floor = floor
        self.num = num
        self.wear = 0
        self.direction = None

    def __str__(self):
        return "Elevator {} on floor {}".format(self.num, self.

    def __repr__(self):
        return "<Elevator object {} currently on floor {}>".for
                                    self.num, self.floo

    def open(self):
        print ("Doors opening on elevator {}.".format(self.num)

    def close(self):
        print ("Doors closing on elevator {}.".format(self.num)

    def move(self, system):
        """Takes an ElevatorSystem and moves one floor in self.

        system.floors[self.floor].remove(self)
        self.wear += 1
        self.floor += self.direction
        system.floors[self.floor].append(self)

        print("On floor {}...".format(self.floor))

    def deactivate(self, system, warning=True):
        """Takes Elevator System to remove self from."""
```

Is the system ready for delivery?

- Email alerts for instant access to the system



plagiarismnotifications@gm...

9:08 PM

Plagiarism Alerts

To: sandip Samal Cc: & 2 more

Hi Instructor,

One or more recent submissions might have exceeded the plagiarism threshold limit.
Kindly view this link :http://ec2-54-145-246-162.compute-1.amazonaws.com:8000/#/instructor/4 in to your system.

Best,
Team-211

- Maintains multiple submissions of a single student
- Runs on AWS, just Browse and Use

Future Scope

- System can be enabled to span across same courses taught by different instructors and previous semesters
- Email alerts can be shared among other course faculties
- Students can submit folder/ path to GitHub to make a submission
- Application can be easily made mobile/tablet compatible

Thank you
Sherlock
Team-211