PHASE B REPORT

CS 5500: Foundations of Software Engineering
Instructor: Professor Bell

Archit Manek, Shebna Mathew, Devina Raithatha, and Nikhita Singh

October 16, 2020

REQUIREMENTS

No.	High Level Functional Requirements	Requirement ID
1	The tool *must* work on code written in at least one language of your choice.	REQ – 01
2	The tool *must* take two programs.	REQ – 02
3	The tool *must* go beyond textual 'diff'.	REQ – 03
4	The tool *must* handle multi-file programs (*should* be able to handle files that have been renamed).	REQ – 04
5	The tool *should* detect strategies to avoid detection such as renaming variables, extracting code into functions, moving code, changing comments etc.	REQ – 05
6	The tool *must* produce an interactive user interface.	REQ – 06
7	The tool *must* use a web-based interface.	REQ – 07
8	The user interface *should* include kind of visual diff of the code portions.	REQ – 08
9	The tool *should* allow a user to selectively examine and compare key portions of the program in the browser.	REQ – 09
10	The tool *should* have an Instructor Login.	REQ – 10
11	The tool *should* be able to take two modes of input: Text/ Upload Files.	REQ – 11
12	The tool *should* display the overall percentage of similarity between two programs as a high-level report.	REQ – 12
13	The tool *should* be able to download the Plagiarism Analysis Report.	REQ – 13
14	The tool *should* offer an option to Log out.	REQ – 14

USE CASES

No.	1
Name	Signing up as a new user
Actor	Instructor
Scenario	1. The Instructor loads the Plagiarism Detector web application. 2. The system displays the Login page with the Sign In fields. 3. A 'New User? Create an account.' link is displayed above it. 4. The Instructor clicks on the 'Create an account' link. 5. The system displays a new page with a Sign-Up form. 6. The Instructor signs up with the username as the school email ID, a password, first name and last name. 7. The Instructor clicks on the 'Sign Up' button to sign up. 8. The system displays if the Registration was successful or if the user is already registered. 9. A link to go back to login page is displayed below it.
Exceptions	The username is not a school email ID
Priority	Should have
Requirement ID	REQ-06, REQ-07, REQ-10

No.	2
Name	Signing into the web application
Actor	Instructor
Scenario	1. The Instructor loads the Plagiarism Detector web application. 2. The system displays the Login page. 3. The Instructor fills in their username and password. 4. The Instructor clicks on the 'Sign In' button. 5. The system loads the tool homepage.
Exceptions	Login fails – User not registered
Priority	Should have
Requirement ID	REQ-06, REQ-07, REQ-10

No.	3
Name	Selecting the Programming language
Actor	Instructor
Scenario	1. The Instructor logs in and loads the tool homepage which displays two panes on each side of the page. - One pane for each program – each containing two buttons for either Text input or Upload Files. - A drop-down menu on top indicates the selected programming language. - There is at least one language in the menu. - No language is defaulted.

	2. The Instructor selects a language.
Exceptions	
Priority	Must have
Requirement ID	REQ-01, REQ-06, REQ-07, REQ-11

No.	4
Name	Inputting the source code for comparison
Actor	Instructor
Scenario	1. On the tool homepage, the Instructor either enters the two programs as plain text or uploads them as files (multiple files for each program allowed). - A 'File Upload' button is displayed at the bottom. 2. The Instructor clicks 'File Upload'.
Exceptions	The input code does not match the selected language.
Priority	Must have
Requirement ID	REQ-02, REQ-04, REQ-06, REQ-11

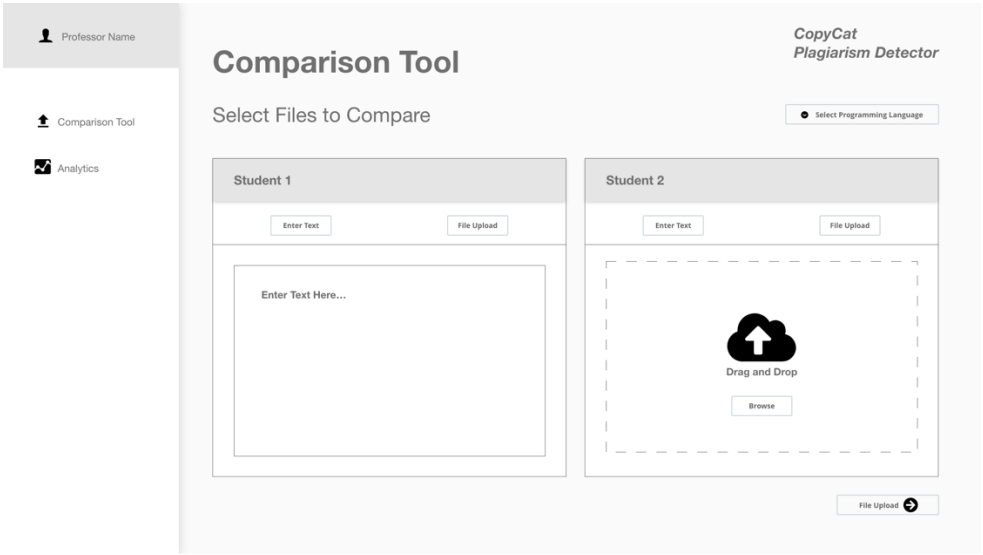
No.	5
Name	Instructor compares source code to find similarities
Actor	Instructor
Scenario	1. The Instructor logs in and loads the tool homepage. 2. The Instructor either enters the two programs as plain text or uploads them as files (multiple files for each program allowed). 3. The Instructor clicks on the 'File Upload' button after inputting the code, which takes them to the 'Analytics' tab. 4. There are 5 sections on the page: a. Filter criteria, which is currently expanded b. Match Percentage c. Similarity Overview d. Selections for Review e. Download Report 5. The Instructor chooses from the options in the Filter Criteria to customize the comparison e.g. Ignore whitespaces, ignore comments, ignore Function names, etc. 6. A 'Compare' button is displayed below the criteria. 7. The Instructor clicks on 'Compare'. 8. The 'Match Percentage' expands and shows the overall percentage of code match and a high-level similarity analysis. 9. The 'Similarity Overview' expands and displays two panes with each program and the sections of code that match as highlighted in the same color.
Exceptions	
Priority	Must have
Requirement ID	REQ-03, REQ-05, REQ-08, REQ-12

No.	6
Name	Selectively Compare code
Actor	Instructor
Scenario	1. The Instructor logs in and loads the tool homepage. 2. The Instructor either enters the two programs as plain text or uploads them as files (multiple files for each program allowed). 3. When the Instructor clicks the 'Compare' button, the 'Similarity Overview' displays the two programs on two panes with the sections of code that match as highlighted with the same color. 4. The Instructor selects any highlighted section (if any) to selectively examine it. 5. The 'Selection for Review' section expands and displays the selected code and explains the similarity in more detail.
Exceptions	
Priority	Should have
Requirement ID	REQ-09

No.	7
Name	Download the Analysis report
Actor	Instructor
Scenario	1. The Instructor logs in and loads the tool homepage. 2. The Instructor either enters the two programs as plain text or uploads them as files (multiple files for each program allowed). 3. The Instructor runs a code comparison by clicking the 'Compare' button on the Analytics tab. 4. Expand the 'Download Report' section to download the analysis report.
Exceptions	
Priority	Should Have
Requirement ID	REQ-13

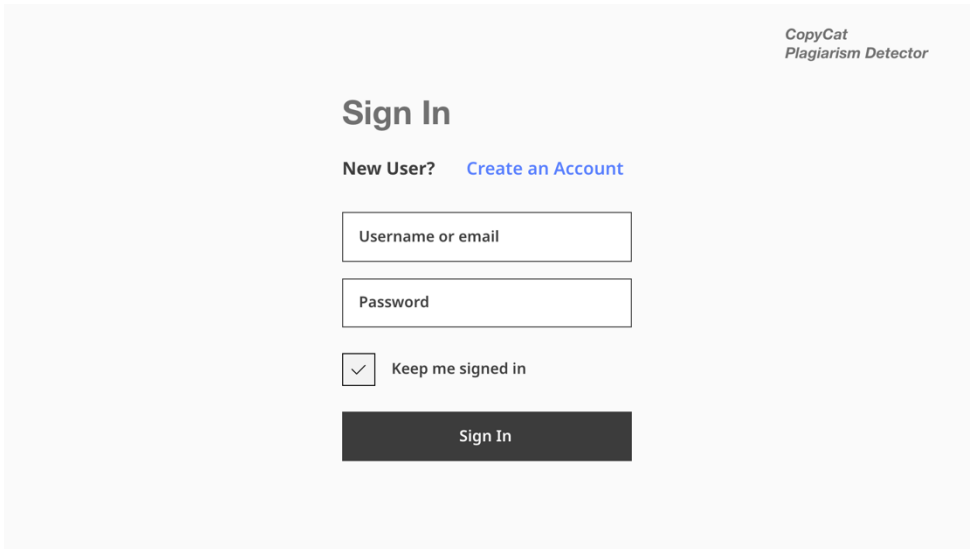
No.	8
Name	Logging out of the web application
Actor	Student/Instructor
Scenario	1. The Student/Instructor clicks on the Profile icon. 2. A 'Log Out' option displays below it. 3. The Student/Instructor clicks on 'Log Out'. 4. The Login page is displayed.
Exceptions	
Priority	Should have
Requirement ID	REQ-14

USER INTERFACE (UI) MOCK-UPS



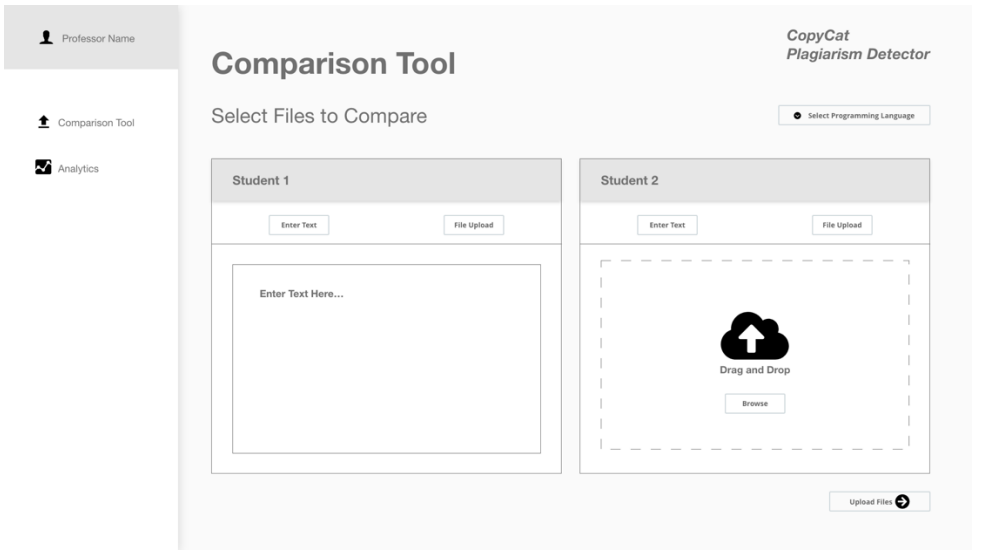
This mockup represents a sign-up page for the plagiarism detector. There is a sign-in button for existing users. Users must input the first name and last name to create an account.

Use Case 1



This mockup represents a sign-in page for the plagiarism detector. There is a sign-up button for new users that want to create an account.

Use Case 2

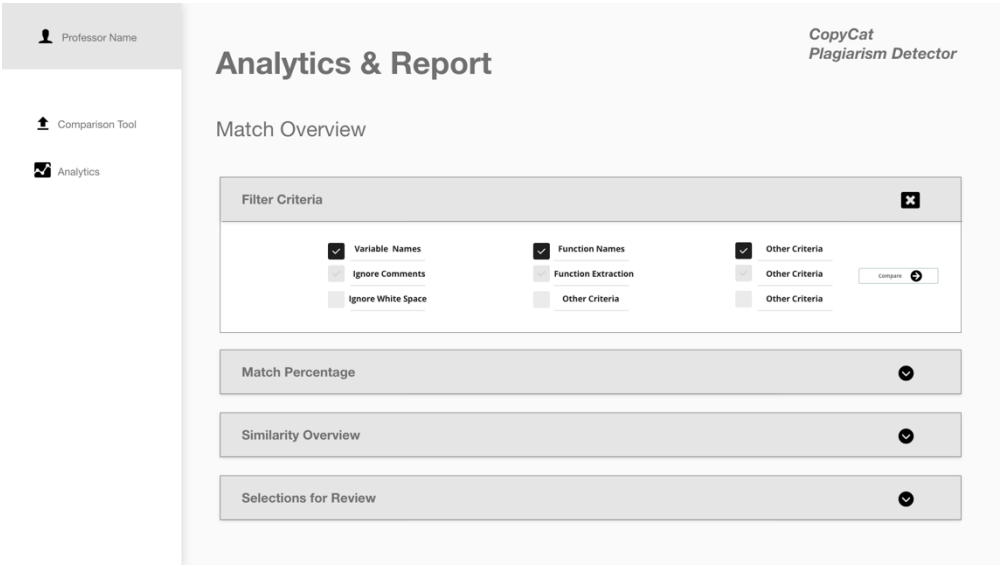


This mockup represents the comparison tool and displays the two ways that a file can be uploaded. The instructor can either insert text or upload a file. To upload a file, there are two options, one is browsing and the other is drag and drop. The instructor can select a language in the dropdown menu on the top right, and the instructor must also select upload files to upload the files successfully. Upload files also works for when the instructor types the text, as this will successfully upload any input to the detector. In addition, the instructor can use the tools on the left menu to navigate the website.

Use Case 3, Use Case 4

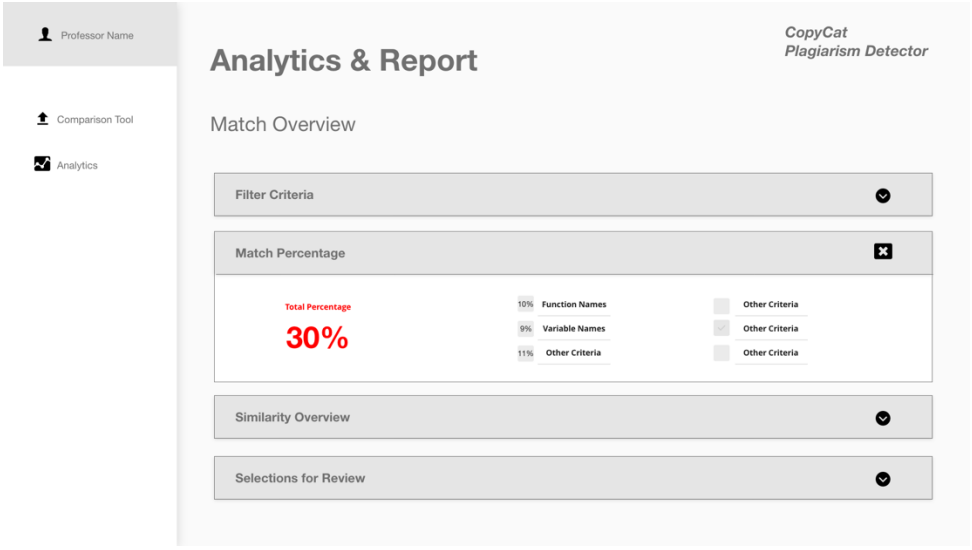
This mockup represents the analytics page with all the various drop-down and reporting options that are available to the instructor.

Use Case 5, 6, 7



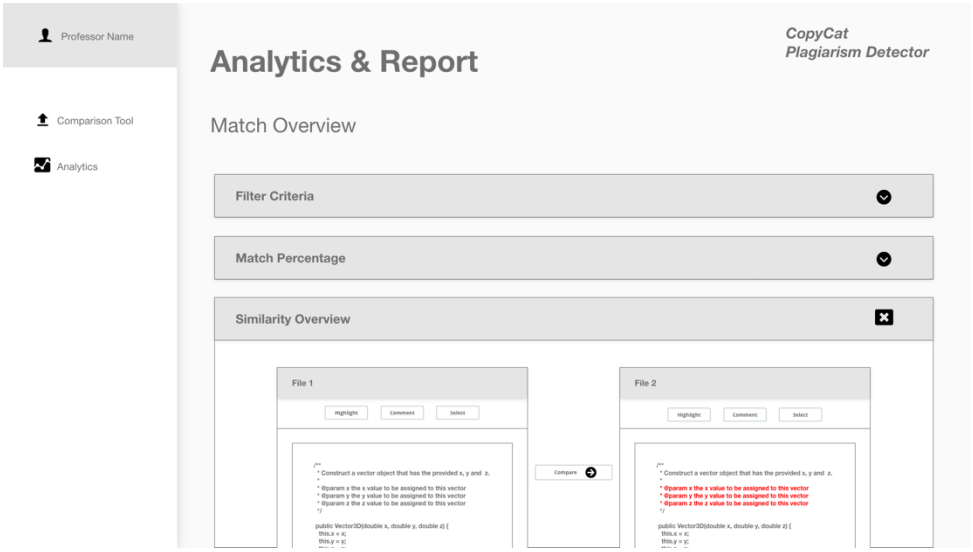
This mockup represents the various criteria options that can be used to compare the files, the instructor can either enter criteria, or use the pre-defined criteria. The instructor can select compare here. We have not determined how they will be able to add their own criteria, but we wanted to include this functionality and display that there is the ability to control which components of the two files are compared against each other.

Use Case 5



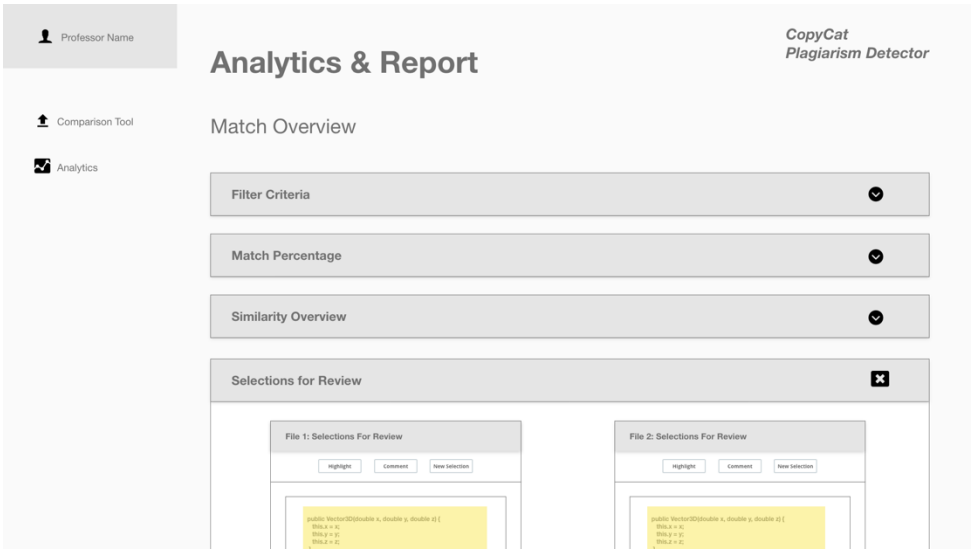
This mockup represents the match percentage between the two files, this is a calculated overview and the breakdown is displayed to the right based on the selected criteria. In addition, the criteria highlight the similarity percentage, therefore this gives the instructor insight into where the comparison total match percentage is coming from.

Use Case 5



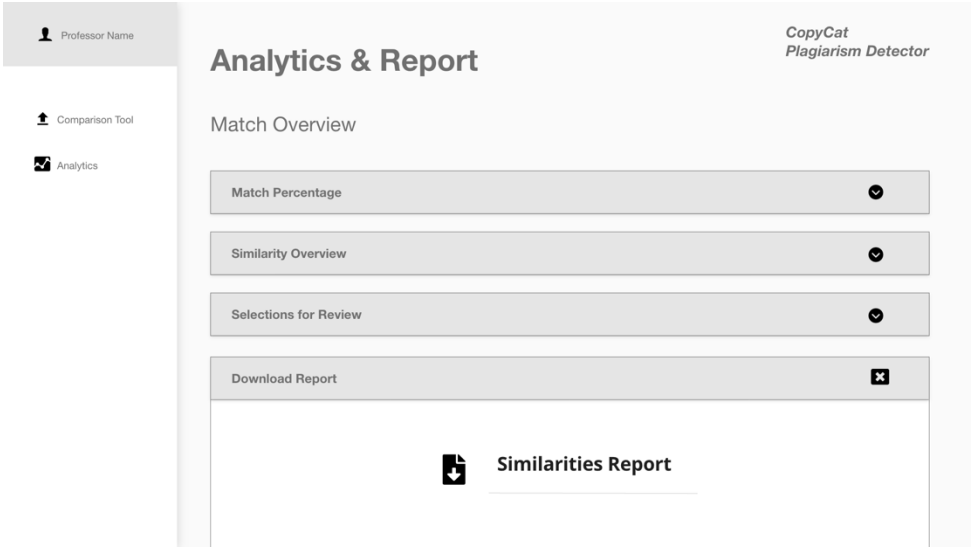
This mockup represents and shows the original files side by side. There are three tools that an instructor can use, highlight, comment, and select. Upon selection, the instructor can compare selectively by pressing the compare button in the center which will then allow for selective comparison.

Use Case 5



This mockup represents and shows the selections that have been selected for review, or to selectively examine a segment of the files. These will be highlighted in yellow to help the instructor visualize the comparisons and similarities and distinguish from the previous tab.

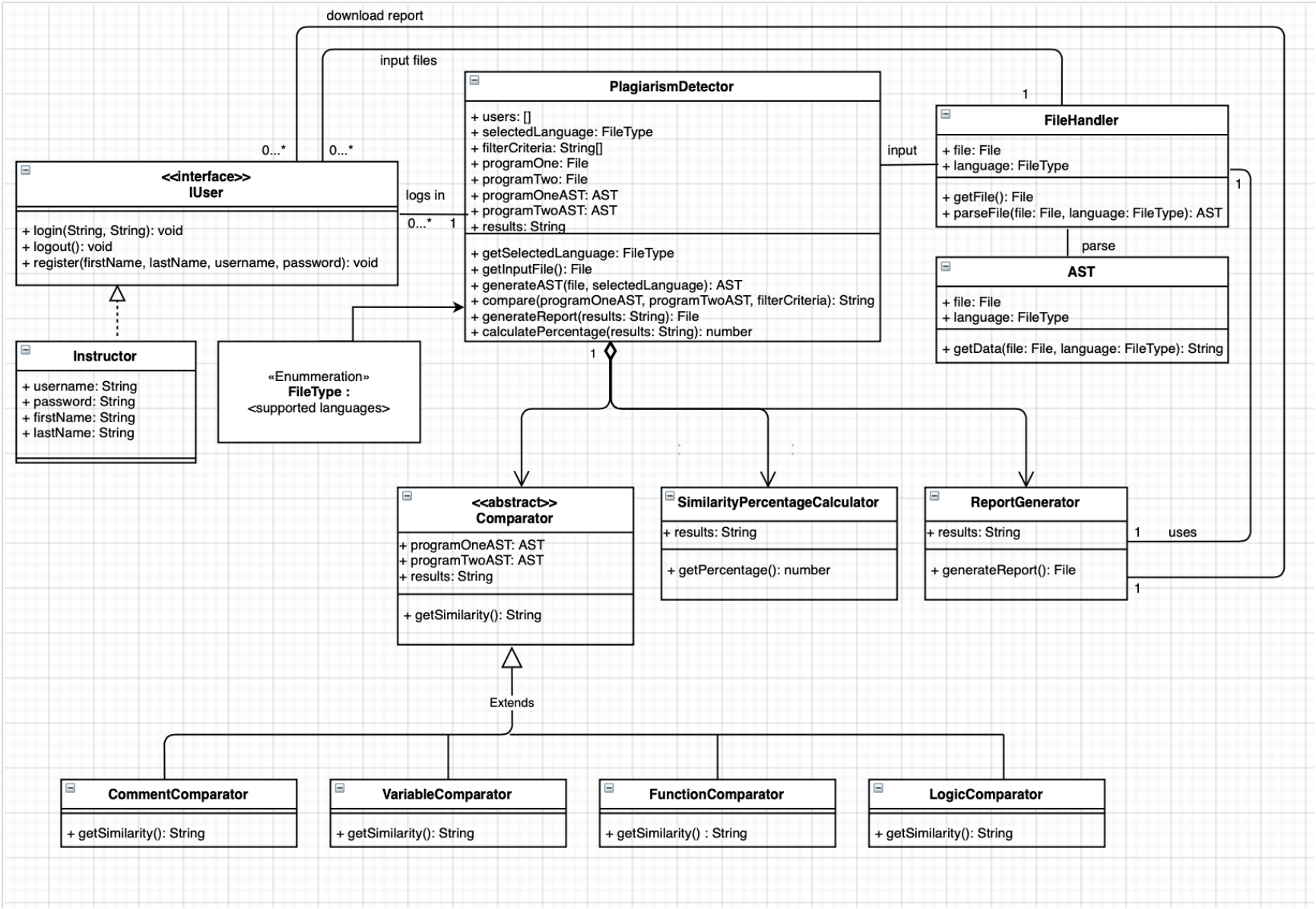
Use Case 6



This mockup represents and shows the download report button that the instructor will be able to download which will summarize the similarity overview for an instructor.

Use Case 7

UML


download report
input files
0...*
0...*
<<interface>>
IUser
+ login(String, String): void
+ logout(): void
+ register(firstName, lastName, username, password): void
logs in
0...*
1
PlagiarismDetector
+ users: []
+ selectedLanguage: FileType
+ filterCriteria: String[]
+ programOne: File
+ programTwo: File
+ programOneAST: AST
+ programTwoAST: AST
+ results: String
+ getSelectedLanguage: FileType
+ getInputFile(): File
+ generateAST(file, selectedLanguage): AST
+ compare(programOneAST, programTwoAST, filterCriteria): String
+ generateReport(results: String): File
+ calculatePercentage(results: String): number
input
1
FileHandler
+ file: File
+ language: FileType
+ getFile(): File
+ parseFile(file: File, language: FileType): AST
1
parse
AST
+ file: File
+ language: FileType
+ getData(file: File, language: FileType): String
Instructor
+ username: String
+ password: String
+ firstName: String
+ lastName: String
«Enummeration»
FileType :
<supported languages>
1
<<abstract>>
Comparator
+ programOneAST: AST
+ programTwoAST: AST
+ results: String
+ getSimilarity(): String
SimilarityPercentageCalculator
+ results: String
+ getPercentage(): number
ReportGenerator
+ results: String
+ generateReport(): File
1
uses
1
Extends
CommentComparator
+ getSimilarity(): String
VariableComparator
+ getSimilarity(): String
FunctionComparator
+ getSimilarity() : String
LogicComparator
+ getSimilarity(): String

AST.ts

phaseB > src > AST.ts > ...

```ts
Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
import { FileType } from "./FileType"      Nikhita Singh, 8 hours ago • Changes to UML and TS code

Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
class AST {
  private file: File
  private language: FileType

  getData(file: File, language: FileType): String {
    throw new console.error("Method not implemented");
  }
}

export default AST
```

CommentComparator.ts

phaseB > src > CommentComparator.ts > ...

```ts
Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
import Comparator from "./Comparator";      Nikhita Singh, a day ago • Phase B TS

Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
class CommentComparator extends Comparator{

  getSimilarity(): String{
    throw new console.error("Method not implemented");
  }
}
```

Comparator.ts

phaseB > src > Comparator.ts > Comparator

```ts
Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
abstract class Comparator {      Nikhita Singh, a day ago • Phase B TS

  protected programOne: File
  protected programTwo: File
  protected results: String

  abstract getSimilarity(): String

}

export default Comparator
```

FileHandler.ts

phaseB > src > FileHandler.ts > ...

```ts
Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
import AST from "./AST";      Nikhita Singh, a day ago • Phase B TS Code
import { FileType } from "./FileType";

Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
class FileHandler {
  private file: File
  private language: FileType

  getFile(): File {
    throw new console.error("Method not implemented");
  }
  parseFile(file: File, language: FileType): AST {
    throw new console.error("Method not implemented");
  }
}
```

FileType.ts

phaseB > src > FileType.ts > FileType

```ts
Nikhita Singh, a day ago | 1 author (Nikhita Singh)
export enum FileType {      Nikhita Singh, a day ago • Phase B TS
  python,
  java,
  typescript
}
```

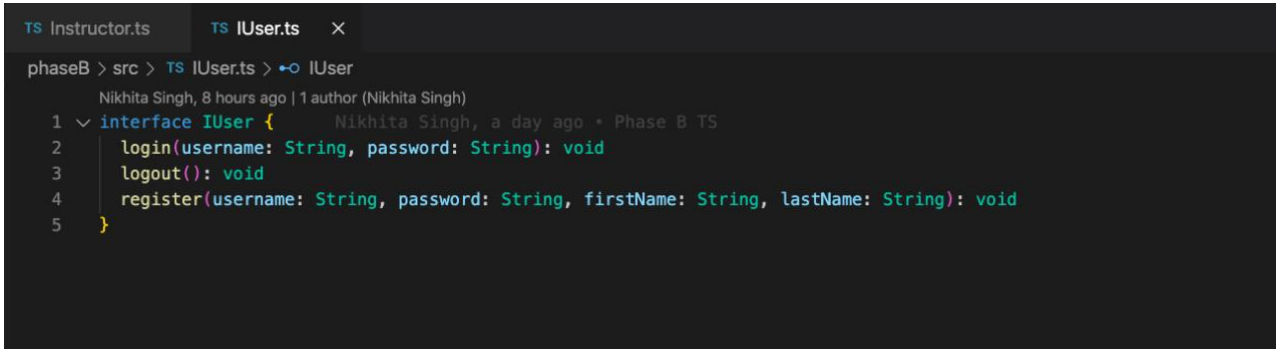
FunctionComparator.ts

phaseB > src > FunctionComparator.ts > ...

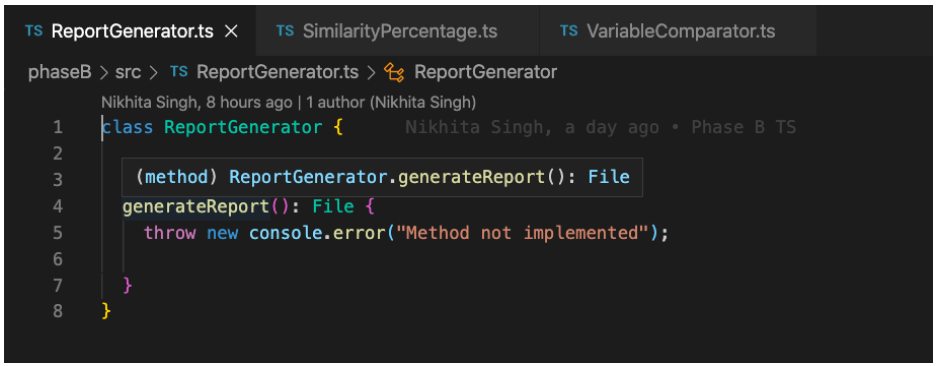
```ts
Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
import Comparator from "./Comparator";      Nikhita Singh, a day ago • Phase B TS

Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
class FunctionComparator extends Comparator {
  getSimilarity(): String {
    throw new Error("Method not implemented.");
  }
}
```

-


Instructor.ts
IUser.ts
phaseB > src > Instructor.ts > Instructor
Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
class Instructor implements IUser {
Nikhita Singh, a day ago • Phase B TS
private username: String
private password: String
private firstName: String
private lastName: String
login(username: String, password: String): void {
throw new Error("Method not implemented.");
}
logout(): void {
throw new Error("Method not implemented.");
}
register(username: String, password: String, firstName: String, lastName: String): void {
throw new Error("Method not implemented.");
}
}


Instructor.ts
IUser.ts
phaseB > src > IUser.ts > IUser
Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
interface IUser {
Nikhita Singh, a day ago • Phase B TS
login(username: String, password: String): void
logout(): void
register(username: String, password: String, firstName: String, lastName: String): void
}


LogicComparator.ts
PlagiarismDetector.ts
ReportGenerator.ts
SimilarityPercentage.ts
phaseB > src > LogicComparator.ts > ...
Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
import Comparator from "./Comparator";
Nikhita Singh, a day ago • Phase B TS
Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
class LogicComparator extends Comparator {
getSimilarity(): String {
throw new Error("Method not implemented.");
}
}


ReportGenerator.ts
SimilarityPercentage.ts
VariableComparator.ts
phaseB > src > ReportGenerator.ts > ReportGenerator
Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
class ReportGenerator {
Nikhita Singh, a day ago • Phase B TS
(method) ReportGenerator.generateReport(): File
generateReport(): File {
throw new console.error("Method not implemented");
}
}

TS PlagiarismDetector.ts

phaseB > src > TS PlagiarismDetector.ts > ...

```ts
import { FileType } from "./FileType";      Nikhita Singh, a day ago • Phase B TS
import AST from './AST';

Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
class PlagiarismDetector {
  private users: String[]
  private selectedLanguage: FileType
  private filterCriteria: String[]
  private programOneAST: AST
  private programTwoAST: AST
  private programOne: File
  private programTwo: File
  private results: String

  getSelectedLanguage(): FileType {
    throw new console.error("Method not implemented");
  }

  getInputFile(): File {
    throw new console.error("Method not implments");
  }

  generateAST(file: File, language: FileType): AST {
    throw new console.error("Method not implemented");
  }

  compare(programOneAST: AST, programTwoAST: AST, filterCriteria: String[]): String {
    throw new console.error("Method not implemented");
  }

  generateReport(results: String): File {
    throw new console.error("Method not implemented");
  }

  calculatePercentage(results: String): number {
    throw new console.error("Method not implemented");
  }
}

export default PlagiarismDetector
```

TS SimilarityPercentage.ts

phaseB > src > TS SimilarityPercentage.ts > SimilarityPercentage

Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)

```ts
class SimilarityPercentage {      Nikhita Singh, a day ago • Phase B TS
  private results: String
  getPercentage(): number {
    throw new console.error("Method not implemented");
  }
}
```

TS VariableComparator.ts

phaseB > src > TS VariableComparator.ts > ...

Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)

```ts
import Comparator from "./Comparator";      Nikhita Singh, a day ago • Phase B TS

Nikhita Singh, 8 hours ago | 1 author (Nikhita Singh)
class VariableComparator extends Comparator {
  getSimilarity(): String {
    throw new Error("Method not implemented.");
  }
}
```