# Principles of Software Engineering Summer 2023

# [FAU: CEN 4010]

# **AI DETECTION TOOL**

# Group 20

# Jordan Wan

# Andrew Ramirez

# Victor Ribas

# Austin Vasquez

# Ruchit Patel

# Milestone 1

# 6/17/2023

# Revision History

# 

# **Table of Contents**

# Executive Summary ……. P. 3

Use Cases……. P. 5

Data Definition…… P. 6

Initial List of Function Specifications…… P. 8

List of Non-Functional Specifications…… P. 10

Competitive Analysis…… P. 13

High-Level System Architecture…… P. 15

Team Roles…… P. 17

# 

# 

# 

# 

# 

# 

# 

# 

# **Executive Summary**

# In the modern digital age, we are often faced with the challenge of differentiating authentic human-generated content from that created by artificial intelligence. AI has evolved significantly, enabling the creation of realistic text, images, and videos, making it difficult to discern between human and AI creations. This is where our AI detection tool, CertifAI, comes into play.

# There are numerous AI detection tools, but they share a common issue - being technologically abundant but accuracy scarce. This implies that despite their sophisticated technologies, if they can't accurately and consistently identify AI-generated content, their utility is limited. The challenge becomes not just detecting AI creations, but accurately identifying them in a diverse array of formats including text, images, and videos. Every individual's interaction with digital content is unique; a robust solution for one person may not work as effectively for another. There is a need for an AI detection tool that takes into account these differences, while simultaneously offering robust, cross-format detection. CertifAI is designed with this very objective in mind.

# CertifAI, named for its core mission of providing "certified AI" detection, was developed with the ambition to become a precise tool for users across the globe. It offers differentiated results that are calibrated to individual needs, uses advanced learning algorithms to continuously improve its detection capabilities, and maintains a dynamic database that evolves with emerging AI trends.

# CertifAI recognizes the diversity of AI-generated content, distinguishing between text, images, and video. It allows users to understand the authenticity of the content they encounter. By understanding what AI models are commonly used, the tool can differentiate between human and AI-generated content more accurately. Users can also contribute to the database, ensuring that it remains comprehensive and up-to-date, reflecting the dynamic and ever-changing nature of AI-generated content.

# With the help of CertifAI, users can plan their interactions with digital content, using our advanced detection capabilities to verify authenticity before engagement. The tool reduces the time spent questioning the authenticity of content by logically storing data and returning results swiftly, while still providing the most accurate detection based on each individual user’s needs.

# In conclusion, CertifAI is a tool that guarantees precision, speed, and convenience in detecting AI-generated content. It is designed to ensure that our interactions with the digital world are genuine and trustworthy. CertifAI - an AI detection tool that’s an internet user's wish come true.

# **AI Detection Tool Proposal**

## **Overview**

The AI detection tool is a cutting-edge tool that detects and prevents AI-generated text, images, and videos. This tool has been developed with the goal of becoming the most reliable and accurate detection method for AI-generated content for organizations everywhere.

## **Responsibilities**

* Develop algorithms to detect AI-generated text, images, and videos
* Collaborate with other researchers and scientists to design and implement the tool
* Analyze data, interpret results, and redact findings for publication
* Develop a commercial product for organizations to use

## **Qualifications**

* Research in relevant field
* Proven track record of developing algorithms and publishing findings
* Demonstrated ability to work collaboratively as part of a research team
* Experience in developing commercial products

## **Milestones**

1. Develop algorithms to detect AI-generated text
2. Develop algorithms to detect AI-generated images
3. Develop algorithms to detect AI-generated videos
4. Test and refine algorithms
5. Publish findings in academic journals and relevant conferences
6. Develop a commercial product for organizations to use

## **Use Cases**

### **Use Case - Detection**

The user will upload a text, image, or video file to the AI detection tool. The tool will analyze the file and determine if it was generated by AI. The tool will provide a confidence score to indicate the likelihood of the file being AI-generated.

1. **Description:** This use case describes the process of how the user will utilize the detection feature of the AI detection tool.
2. **Actors:** 2.1 User 2.2 System
3. **Preconditions:** 3.1 User has an active internet connection 3.2 System is available
4. **Primary Flow of Events:**
5. User uploads a text, image, or video file.
6. System analyzes the file.
7. System determines if the file was generated by AI.
8. System provides a confidence score to indicate the likelihood of the file being AI-generated.
9. Terminate Use Case: Detection
10. **Alternate Flows** 5.1 **User Uploads Unsupported File Type** If in step 1, user uploads a file type that is not supported by the system.
11. System notifies the user that the file type is not supported.
12. Return to step 1.

5.2 **System is Unable to Determine if File is AI-Generated** If in step 3, the system is unable to determine if the file was generated by AI.

1. System notifies the user that it was unable to make a determination.
2. Terminate Use Case: Detection

### **Use Case - Analysis**

The user will upload a text, image, or video file to the AI detection tool. The tool will analyze the file and provide detailed analysis on the AI-generated aspects of the file.

1. **Description:** This use case describes the process of how the user will utilize the analysis feature of the AI detection tool.
2. **Actors:** 2.1 User 2.2 System
3. **Preconditions:** 3.1 User has an active internet connection 3.2 System is available
4. **Primary Flow of Events:**
5. User uploads a text, image, or video file.
6. System analyzes the file.
7. System provides detailed analysis on the AI-generated aspects of the file.
8. Terminate Use Case: Analysis
9. **Alternate Flows** 5.1 **User Uploads Unsupported File Type** If in step 1, user uploads a file type that is not supported by the system.
10. System notifies the user that the file type is not supported.
11. Return to step 1.

5.2 **System is Unable to Provide Detailed Analysis** If in step 3, the system is unable to provide detailed analysis on the AI-generated aspects of the file.

1. System notifies the user that it was unable to provide detailed analysis.
2. Terminate Use Case: Analysis

## **Data Definition**

**Name Meaning**

| **Name** | **Meaning** | **Usage** | **Comment** |
| --- | --- | --- | --- |
| File | data | Use case scenarios | General encompass of just letting us know that multiple types of files can be observe |
| Text | data | Use case scenarios | A type of file that includes word files, pdfs, and any other text based file |
| Image | data | Use case scenarios | A type of file that include pngs, jpegs and any other types of static image files |
| Video | data | Use case scenarios | A type of file that includes mp4, MOV and any other typ of video file |
| AI-generated | data | Use case scenarios | This is just over encompasses all the different types of files but that are generated by AI |
| File to be analyzed | service | Site user service | The software will allow all the different files that get sent through it to be evaluated |
| User | Actor | Use case scenarios | Anyone using the software for file analysis |
| tool | User interface | User interface | The tool just references the the file analysis capability of the software |
| system | service | Use case scenarios | All front end, back end, AI database that are used for this software, and all the code |
| platform | service | Use case scenarios | This more describes the where the users are going to be interacting with the software |
| certifAI | Domain Name | Use case scenarios |  |
| GitHub |  |  |  |

## **Initial List of Functional Specifications**

**Non-Member expectation**

**1. Creating Account**

* The system shall allow the user to create an account by storing UserID, Password, Date of Birth, First Name, Last name, Location, and answer to security question/phone number.
* The system shall not allow the User to Create an account if the UserID choose by the User already exist in the System’s Database.
* The system shall prevent the user from creating an account if the User’s chosen password does not match the re-enter password field.
* System shall prevent the creation of the user’s account if following fields are not filled: First Name, Last Name, Location, UserID, Password, Re-enter Password, Security Answer Security Question or Phone number, and Date of Birth.

**2. Detection**

* The user will upload a text, image, or video file to the AI detection tool.
* The tool will analyze the file and determine if it was generated by AI.
* The tool will provide a confidence score to indicate the likelihood of the file being AI-generated.

**3. Analysis**

* The user will upload a text, image, or video file to the AI detection tool.
* The tool will analyze the file and provide detailed analysis on the AI-generated aspects of the file.

**4. Reporting**

* The tool will generate reports that summarize the analysis of AI-generated content for organizations to use.

**5. Integration**

* The tool will integrate with other systems and platforms to provide seamless detection and prevention of AI-generated content.

## List of non-functional specifications for AI Detection Tool:

## 

Performance Requirements:

1. Responsiveness: The AI Detection Tool will be responsive and compatible with various monitor sizes, including 10" netbooks to 24" desktop monitors. It will adapt to different resolutions, ranging from 1024 x 600 through 1900 x 1200.

2. Cycle Time: The tool's cycle time will range from 1.0 to 1.7, depending on the number of concurrent users. It will operate smoothly with 5-50 concurrent users, with increasing lag as the user load exceeds 50. Performance may briefly halt until a user finishes when the number of concurrent users surpasses 50.

3. Speed Per Transaction: The tool will process transactions with a speed ranging from 20 to 100 milliseconds, depending on the cycle time. It can handle 10-50 transactions per second.

4. Test Requirements: Performance testing will include load tests and assessment of the tool's speed per transaction, along with comprehensive testing of all functional specifications.

5. Reliability: The tool should have a mean time between failures of 1 hour or less within a total of 3 months. Maintenance and updates can be performed during this downtime. It should be operational for 99.8% of the calendar year, with a maximum downtime of 0.2%. The system must be operational 100% of the time during the first year of operation.

6. Minimum Bug Counts:

- No more than 5 bugs should be present during integration and testing.

- After delivery, no more than 3 bugs should remain in the system.

7. Execution Speed: The initial home page of the tool should load within 100-200 milliseconds on a high-speed internet connection, considering the current cycle time.

8. Storage Utilization: The tool should utilize storage within the range of 75-90% of the available storage capacity to avoid technical issues. Additional storage should be readily available for emergency situations.

9. Robustness: The tool should restart within an hour after a failure. The probability of events causing failure should be under 0.1%, and the likelihood of data corruption on failure should be below 0.8%.

## Ease of Use:

1. Training Time: The AI Detection Tool should require minimal to no training. It should be user-friendly and accessible to all visitors.

## Interoperability Requirements:

1. Browser Compatibility: The web-based AI Detection Tool should operate on major browsers, including Google Chrome, Mozilla Firefox, Safari, Opera, and Internet Explorer. Alternatives or fallback options should be provided for browsers without JavaScript support.

2. Computer and OS Compatibility: The tool should operate on various operating systems, including Windows, OS X, and Linux. It should also be compatible with any computer capable of running a supported browser.

## Expected Load:

1. The AI Detection Tool should be able to accommodate up to 50 simultaneous users. Load testing will be conducted to measure performance during periods of high traffic, both continuous and spiked.

## Security Requirements:

1. User Login/Password System: The tool should incorporate a secure login/password system to manage user preferences, ratings, and reviews. Password confirmation should be required during account creation. A security question and answer should be stored for password retrieval purposes.

2. Encryption: As the tool does not involve purchases or exchange of valuable information, encryption is not required.

3. Access Control: The development team will have access to edit the front end code, back end code, and databases. Users and visitors will have limited access based on the user interface.

4. Spam Protection: The tool should implement a mechanism to prevent spamming and the creation of bogus accounts, such as requiring users to enter characters displayed in a picture during account creation.

5. Resource Utilization: Resources, including the MySQL database on the sfsuswe.com server, should be accessed using proper usernames and passwords. The tool will utilize frameworks such as Bootstrap and jQuery, ensuring compliance with their licenses and qualifications.

## Portability Requirements:

1. Platform Compatibility: The AI Detection Tool has plans to be developed for mobile and tablet platforms, subject to feasibility within the given time frame. Future versions may include mobile and tablet compatibility.

2. Percentage of Target-Dependent Statements: In this version of the tool, no statements will be target dependent. For future versions designed for mobile and tablet use, 30% of statements may become target dependent.

## Supportability Requirements:

1. Coding Standards: The tool will adhere to coding standards of 75-80% for HTML5 and CSS3. The code will undergo production, review, testing, additional review, and finalization by a dedicated developer for efficiency.

2. Naming Conventions: HTML classes and ID tags will be coded in lowercase, except when multiple words are used, in which case camel case convention will be followed. SQL tables and data names will be agreed upon by the team and use capitalization for the first letter and lowercase for the remaining letters.

## Storage Requirements:

1. The AI Detection Tool will utilize the sfsuswe.com server to store MySQL databases and site files, with the exact capacity yet to be determined.

## Survivability:

1. The system and major files/documentation will be stored on sfsuswe.com servers, with backups maintained on Google Drive to prevent loss in case of server damage or destruction.

## Availability Requirements:

1. Accessible Times: The AI Detection Tool should be available for use 24/7, as long as the sfsuswe.com server remains operational.

2. Downtime Impact: Downtime will be minimal, and a splash page will be used to indicate maintenance periods. Scheduled downtime will be announced in advance, and its impact is expected to be minimal.

3. Support: Support will be available via email, with a responsive time of within 24 hours.

## Fault Tolerance:

1. Exception Handling: The AI Detection Tool will include exception handling to provide users with explanations for exceptions and the opportunity to rectify the situation or return to the home page.

2. Self-checking software will not be implemented, as it is not deemed critical for this system.

## **Competitive Analysis**

The analysis of competitors' websites will focus on six main features: Homepage, Design, Navigation, Usability, Plagiarism checking, AI text highlighting. The competitive analysis will utilize a numerical scale to rate each site (1=bad, 2=poor, 3=fair, 4=good, 5=outstanding) The websites chosen for this analysis were selected due to their focus on AI writing detection.

|  | **CopyLeaks** | **Ai Text Classifier** | **GPTZero** | **Writer** | **CrossPlag** |
| --- | --- | --- | --- | --- | --- |
| **Homepage:** | **5** | **3** | **4** | **3** | **4** |
| **Design:** | **4** | **2** | **2** | **3** | **4** |
| **Navigation:** | **3** | **2** | **5** | **5** | **4** |
| **Usability:** | **3** | **3** | **2** | **1** | **3** |
| **Plagiarism Checking:** | **3** | **1** | **2** | **1** | **4** |
| **AI text highlighting (AI vs Human text):** | **5** | **1** | **1** | **1** | **2** |
| **Mean:** | **3.8** | **2.0** | **2.7** | **2.3** | **3.5** |

**CopyLeaks (3.8)**<https://copyleaks.com/ai-content-detector>

The CopyLeaks Homepage was informative and was designed very stylishly. It was easy to navigate with the navigation bar at the top of the page, and the interface was user friendly. The plagiarism checking was outstanding and differentiated between human and AI generated text. No registration was required to get started and it even provided a chrome extension.

**Ai Text Classifier (1.7)**<https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text>

AI Text Classifiers’ homepage provided a lot of information on the tool’s limitations, but the design was bland, and the navigation was spread throughout the page. Although it was user friendly, the plagiarism checking was lackluster and only provided a vague result like “considers the text to be **likely** AI-generated.”. The tool made no effort to make any distinction between what is plagiarized and what isn’t, and worst of all, the site requires registration before you can start using it.

**GPTZero (2.7)**<https://gptzero.me/>

GPTZero’s homepage was simple with not too much information. The design was well formatted and used a tranquil color scheme. The navigation bar is locked on the left side of the page and is well labeled and easy to use. A subscription is needed to view more than 5000 characters, so you may need to make several inputs to check a paper, although the site does offer the ability to upload documents.0 The Plagiarism checking wasn’t very great as it told you only if it was likely or not to be AI generated and it gave the text a perplexity and burstiness score which weren’t useful. It does not differentiate between what is human written or AI written.

**Writer (2.5)**<https://writer.com/ai-content-detector/>

Writer’s homepage was not very informative. The design was very plain and felt like a blog since it was in black and white. For the navigation, all the links were placed at the bottom of the page and were organized well. No signup is required to start using it, however there is a 1500-character limit. Plagiarism checking was one of the worst of the websites I encountered as it did not accurately detect AI generated text and would only give a percentage to indicate whether the text is AI generated.

**CrossPlag (3.5)**<https://app.crossplag.com/individual/detector>

CrossPlag’s homepage has everything you need right on the front page. The design is great and features a navigation bar on the left side of the page which is simple to use. The tool allows the user to check up to 3000 words per check, however registration is required prior to start. Plagiarism checking is much more accurate than the other tools tested on this list, however it will only tell you if the text is mainly written by AI and will not differentiate between what is human written and AI written.

**Planned Advantages:**

Our website will focus on ease of use. Many AI detection websites require registration or are inhibited by some form of pay wall. Our site will be free to use and will not require registration. Registration will allow users to see their past work and pick up where they left off. We also plan to incorporate the ability to upload documents in batches as most of our competitors only allow text to be inputted. This will save users time and effort. We plan to give our website the ability to differentiate between AI and human text by highlighting the different portions of the text which are AI generated. Our users will enjoy using our website because it will be simple, elegant, but powerful.

## **High-level System Architecture**

1. lamp.cse.fau.edu Lamp Server: The Lamp Server with FAU Linux Apache MySQL PHP [LAMP] server, shall be hosting our AI detection tool project during the semester.
2. Trello: The Trello application is the means that the group shall be communicating with each other for the development of the project during the semester.
3. Bugzilla: Bugzilla is a bug tracking system used to track project features, issues and bugs, the group will use this tool during the semester to track the development process for the final project
4. Canvas Announcements: Canvas announcements will be used to check if the instructor or TA have posted any information that was not sent through squirrel mail web application
5. mySQL Database: MySQL database is the database that is being used for the data that will be handled for the project. Users will be adding to the database via the website input function and developers will be managing the data (by either deleting) or adding items from and to the database.
6. Visual Studio Code (IDE): Visual Studio Code is the IDE that the developers will be using to create the code for the website Languages to be used for the development of the website will be the following
   1. HyperText Mark-up Language(HTML) - will be the language that will allow the browser display the website
   2. Cascading Style Sheets(CSS) - will be the language used to decorate the web pages
   3. Personal Home Page (PHP) - will be the language used for server side functionality for the database and real time edits in the tables
   4. Javascript - will be the language used for client side functionality that will be handled for User Interface(UI) needs to make the user experience enjoyable
   5. JQuery - will be the language used for client side functionality (Link to License: https://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt) — ( Jquery APIs)
   6. JQuery UI - will be the language used for client side functionality that will be handled for User Interface(UI) needs to make the user experience enjoyable (Link to License: <https://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt>) — Note (Jquery APIs)
   7. Bootstrap - Bootstrap will be the framework that will be used for code construction for web pages within the groups project. (Link to License: <https://github.com/twbs/bootstrap/blob/master/LICENSE>)
7. GitHub: is a tool that the developers will be using to store, share, and allow access for team members to view source code that they have written. It allows for easier version control of the project.
8. Browser Compatibility: The system will be a web-based web app that operates on at least two of the all of the major browsers, including Google Chrome, Mozilla Firefox, Safari, Opera, and Internet Explorer. It will have functionality in it that will provide alternatives if the browser does not have JavaScript installed on it.

## **Check list**

<https://trello.com/invite/b/yu2vTuVR/ATTI53d17b5a8193e3796a95e217f045cf5c24F81071/certifai>

**Github Link:** <https://github.com/KinggWan/AIDetectionTool>

## **Conclusion**

The AI detection tool is a powerful tool that will help organizations detect and prevent AI-generated text, images, and videos. With the development of accurate detection algorithms, and the ability to provide detailed analysis and reporting, the AI detection tool is a valuable asset for organizations everywhere.

**Team Roles:**

**Scrum Master:**

* Jordan Wan

**Product Owner:**

* Victor Ribas

**Front End Developers:**

* Andrew Ramirez

**Back End Developers:**

* Ruchit Patel
* Austin Vasquez