

# Software Requirements Specification for Software Engineering: Code Plagiarism Detector

Team 2, SyntaxSentinals  
Mohammad Mohsin Khan  
Lucas Chen  
Dennis Fong  
Julian Cecchini  
Luigi Quattrociochi

September 29, 2024

# Contents

<b>1</b>	<b>Purpose of the Project</b>	<b>vi</b>
1.1	User Business . . . . .	vi
1.2	Goals of the Project . . . . .	vi
<b>2</b>	<b>Stakeholders</b>	<b>vi</b>
2.1	Client . . . . .	vi
2.2	Customer . . . . .	vi
2.3	Other Stakeholders . . . . .	vi
2.4	Hands-On Users of the Project . . . . .	vi
2.5	Personas . . . . .	vi
2.6	Priorities Assigned to Users . . . . .	vi
2.7	User Participation . . . . .	vii
2.8	Maintenance Users and Service Technicians . . . . .	vii
<b>3</b>	<b>Mandated Constraints</b>	<b>vii</b>
3.1	Solution Constraints . . . . .	vii
3.2	Implementation Environment of the Current System . . . . .	vii
3.3	Partner or Collaborative Applications . . . . .	vii
3.4	Off-the-Shelf Software . . . . .	vii
3.5	Anticipated Workplace Environment . . . . .	vii
3.6	Schedule Constraints . . . . .	vii
3.7	Budget Constraints . . . . .	vii
3.8	Enterprise Constraints . . . . .	viii
<b>4</b>	<b>Naming Conventions and Terminology</b>	<b>viii</b>
4.1	Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project . . . . .	viii
<b>5</b>	<b>Relevant Facts And Assumptions</b>	<b>viii</b>
5.1	Relevant Facts . . . . .	viii
5.2	Business Rules . . . . .	viii
5.3	Assumptions . . . . .	viii
<b>6</b>	<b>The Scope of the Work</b>	<b>viii</b>
6.1	The Current Situation . . . . .	viii
6.2	The Context of the Work . . . . .	ix
6.3	Work Partitioning . . . . .	ix

6.4	Specifying a Business Use Case (BUC)	x
<b>7</b>	<b>Business Data Model and Data Dictionary</b>	<b>x</b>
7.1	Business Data Model	x
7.2	Data Dictionary	xi
<b>8</b>	<b>The Scope of the Product</b>	<b>xi</b>
8.1	Product Boundary	xi
8.2	Product Use Case Table	xi
8.3	Individual Product Use Cases (PUC's)	xi
<b>9</b>	<b>Functional Requirements</b>	<b>xi</b>
9.1	Functional Requirements	xi
<b>10</b>	<b>Look and Feel Requirements</b>	<b>xi</b>
10.1	Appearance Requirements	xi
10.2	Style Requirements	xi
<b>11</b>	<b>Usability and Humanity Requirements</b>	<b>xii</b>
11.1	Ease of Use Requirements	xii
11.2	Personalization and Internationalization Requirements	xii
11.3	Learning Requirements	xii
11.4	Understandability and Politeness Requirements	xiii
11.5	Accessibility Requirements	xiii
<b>12</b>	<b>Performance Requirements</b>	<b>xiii</b>
12.1	Speed and Latency Requirements	xiii
12.2	Safety-Critical Requirements	xiii
12.3	Precision or Accuracy Requirements	xiii
12.4	Robustness or Fault-Tolerance Requirements	xiv
12.5	Capacity Requirements	xiv
12.6	Scalability or Extensibility Requirements	xiv
12.7	Longevity Requirements	xiv
<b>13</b>	<b>Operational and Environmental Requirements</b>	<b>xiv</b>
13.1	Expected Physical Environment	xiv
13.2	Wider Environment Requirements	xiv
13.3	Requirements for Interfacing with Adjacent Systems	xiv
13.4	Productization Requirements	xiv

13.5 Release Requirements . . . . .	xv
<b>14 Maintainability and Support Requirements</b>	<b>xv</b>
14.1 Maintenance Requirements . . . . .	xv
14.2 Supportability Requirements . . . . .	xv
14.3 Adaptability Requirements . . . . .	xv
<b>15 Security Requirements</b>	<b>xv</b>
15.1 Access Requirements . . . . .	xv
15.2 Integrity Requirements . . . . .	xv
15.3 Privacy Requirements . . . . .	xv
15.4 Audit Requirements . . . . .	xv
15.5 Immunity Requirements . . . . .	xvi
<b>16 Cultural Requirements</b>	<b>xvi</b>
16.1 Cultural Requirements . . . . .	xvi
<b>17 Compliance Requirements</b>	<b>xvi</b>
17.1 Legal Requirements . . . . .	xvi
17.2 Standards Compliance Requirements . . . . .	xvi
<b>18 Open Issues</b>	<b>xvi</b>
<b>19 Off-the-Shelf Solutions</b>	<b>xvi</b>
19.1 Ready-Made Products . . . . .	xvi
19.2 Reusable Components . . . . .	xvi
19.3 Products That Can Be Copied . . . . .	xvi
<b>20 New Problems</b>	<b>xvii</b>
20.1 Effects on the Current Environment . . . . .	xvii
20.2 Effects on the Installed Systems . . . . .	xvii
20.3 Potential User Problems . . . . .	xvii
20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product . . . . .	xvii
20.5 Follow-Up Problems . . . . .	xvii
<b>21 Tasks</b>	<b>xvii</b>
21.1 Project Planning . . . . .	xvii
21.2 Planning of the Development Phases . . . . .	xvii

<b>22 Migration to the New Product</b>	<b>xviii</b>
22.1 Requirements for Migration to the New Product . . . . .	xviii
22.2 Data That Has to be Modified or Translated for the New System	xviii
<b>23 Costs</b>	<b>xviii</b>
<b>24 User Documentation and Training</b>	<b>xviii</b>
24.1 User Documentation Requirements . . . . .	xviii
24.2 Training Requirements . . . . .	xviii
<b>25 Waiting Room</b>	<b>xviii</b>
<b>26 Ideas for Solution</b>	<b>xviii</b>

## Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

# **1 Purpose of the Project**

## **1.1 User Business**

*Insert your content here.*

## **1.2 Goals of the Project**

*Insert your content here.*

# **2 Stakeholders**

## **2.1 Client**

*Insert your content here.*

## **2.2 Customer**

*Insert your content here.*

## **2.3 Other Stakeholders**

*Insert your content here.*

## **2.4 Hands-On Users of the Project**

*Insert your content here.*

## **2.5 Personas**

*Insert your content here.*

## **2.6 Priorities Assigned to Users**

*Insert your content here.*

## **2.7 User Participation**

*Insert your content here.*

## **2.8 Maintenance Users and Service Technicians**

*Insert your content here.*

# **3 Mandated Constraints**

## **3.1 Solution Constraints**

*Insert your content here.*

## **3.2 Implementation Environment of the Current System**

*Insert your content here.*

## **3.3 Partner or Collaborative Applications**

*Insert your content here.*

## **3.4 Off-the-Shelf Software**

*Insert your content here.*

## **3.5 Anticipated Workplace Environment**

*Insert your content here.*

## **3.6 Schedule Constraints**

*Insert your content here.*

## **3.7 Budget Constraints**

*Insert your content here.*



### **3.8 Enterprise Constraints**

*Insert your content here.*

## **4 Naming Conventions and Terminology**

### **4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project**

*Insert your content here.*

## **5 Relevant Facts And Assumptions**

### **5.1 Relevant Facts**

*Insert your content here.*

### **5.2 Business Rules**

*Insert your content here.*

### **5.3 Assumptions**

*Insert your content here.*

## **6 The Scope of the Work**

### **6.1 The Current Situation**

The current code plagiarism detection tools such as MOSS rely on tokenization and syntax-level comparisons. Although effective in detecting direct copies or slight variations, they struggle when faced with techniques such as adding redundant code which allows the user to completely bypass detection while still plagiarizing the underlying logic and structure of the code.

Additionally, MOSS does not take into account for the complexity or intent behind the code, leading to issues such as false positives for common programming patterns. This creates a gap for more advanced tools capable

of understanding the semantic meaning of code to more accurately detect plagiarism.

## 6.2 The Context of the Work

Our project aims to address these gaps by incorporating Natural Language Processing (NLP) and machine learning techniques which will be leveraged to improve the accuracy of detecting copied code. The context of the work is within academic institutions, where the integrity of student work is aparmount and our tool will be used by professors to ensure a fair grading process while also supporting students in understanding the ethical use of code.

## 6.3 Work Partitioning

- **Research and Design:** Conduct research on current plagiarism detection systems and state-of-the-art NLP techniques applicable to code plagiarism.
- **Data Collection:** Gather a dataset of code snippets, including both plagiarized and original works, to train and test the model.
- **Model Development:** Develop the NLP-based model capable of understanding the semantic meaning of code. This may involve exploring techniques like abstract syntax trees (ASTs), vector embeddings, or other representations of code that retain semantic meaning.
- **System Integration:** Build the system to take code as input, run through the developed model, and output a similarity score with appropriate thresholds.
- **Testing and Validation:** Test the system with various code samples to validate its performance and accuracy compared to traditional systems like MOSS. This will also test wether our method produces any false positives.
- **Documentation and Deployment:** Document the system architecture, the model, and the results. Deploy the system for use within academic settings.

## 6.4 Specifying a Business Use Case (BUC)

**Business Use Case:** Automated Code Plagiarism Detection for Academic Institutions

- **Actors:** Professors, Students, System Administrators
- **Trigger:** A professor or system administrator uploads multiple code submissions for plagiarism detection in a course assignment.

### Main Success Scenario

1. The system ingests the uploaded code submissions.
2. The system processes each code snippet using the NLP model to generate semantic representations of the code.
3. The system compares the representations to detect plagiarism, taking into account code similarity beyond syntax or token matching.
4. The system outputs a similarity score for each comparison, with thresholds indicating whether plagiarism is suspected.
5. The professor reviews the similarity scores and flags any suspicious cases for further investigation.
6. The system generates a report summarizing the findings for the professor's review.

### Extensions

- If the system detects false positives (common programming patterns being flagged as plagiarism), the professor can override the result.
- If new sophisticated plagiarism techniques are detected, the system can update its learning algorithms to improve accuracy over time.

## 7 Business Data Model and Data Dictionary

### 7.1 Business Data Model

*Insert your content here.*

## **7.2 Data Dictionary**

*Insert your content here.*

# **8 The Scope of the Product**

## **8.1 Product Boundary**

*Insert your content here.*

## **8.2 Product Use Case Table**

*Insert your content here.*

## **8.3 Individual Product Use Cases (PUC's)**

*Insert your content here.*

# **9 Functional Requirements**

## **9.1 Functional Requirements**

*Insert your content here.*

# **10 Look and Feel Requirements**

## **10.1 Appearance Requirements**

*Insert your content here.*

## **10.2 Style Requirements**

*Insert your content here.*

## 11 Usability and Humanity Requirements

### 11.1 Ease of Use Requirements

The system should be intuitive and simple to use for the target audience.

- **Minimal Learning Curve:** Users should be able to complete tasks with minimal instruction or training, taking at most 5 minutes from start to finish. The interface must guide users intuitively through this process.
- **Task Efficiency:** Common tasks should be achievable in no more than three clicks or interactions from the main screen.
- **Clear Navigation:** All navigation elements should be labeled clearly and positioned consistently across different pages to avoid confusion.

### 11.2 Personalization and Internationalization Requirements

The system will support a customizable experience in the future, but users will not be able to modify key settings such as themes or layout preferences at launch.

- **Future Customization Support:** Future updates will allow users to modify settings such as themes and layouts to suit their personal preferences.
- **English Language Support:** The system will operate exclusively in English, with all dates, currency, and numeric formats adhering to English (US) standards.

### 11.3 Learning Requirements

The system should provide clear documentation and onboarding materials.

- **Onboarding:** A guided onboarding process should be available for new users, helping them understand the main features of the system in under 5 minutes.

- **Help Documentation:** Detailed help documentation and tooltips should be available for key features to reduce the need for external assistance.

## 11.4 Understandability and Politeness Requirements

The system should use clear language and maintain a polite tone in all interactions with users.

- **Clear Language:** All messages and labels should be in simple, every-day language to ensure clarity for users of different levels of technical expertise.
- **Politeness:** Error messages and prompts should be worded politely and offer constructive guidance. Example error messages include:
  - *“Oops! Something went wrong. Please try again or contact support if the issue persists.”*
  - *“We’re sorry, but the file you uploaded is not supported. Please upload a .py file.”*

## 11.5 Accessibility Requirements

Due to time constraints, the system will not be fully accessible at launch. But this can be a future goal.

# 12 Performance Requirements

## 12.1 Speed and Latency Requirements

*Insert your content here.*

## 12.2 Safety-Critical Requirements

*Insert your content here.*

## 12.3 Precision or Accuracy Requirements

*Insert your content here.*

## **12.4 Robustness or Fault-Tolerance Requirements**

*Insert your content here.*

## **12.5 Capacity Requirements**

*Insert your content here.*

## **12.6 Scalability or Extensibility Requirements**

*Insert your content here.*

## **12.7 Longevity Requirements**

*Insert your content here.*

# **13 Operational and Environmental Requirements**

## **13.1 Expected Physical Environment**

*Insert your content here.*

## **13.2 Wider Environment Requirements**

*Insert your content here.*

## **13.3 Requirements for Interfacing with Adjacent Systems**

*Insert your content here.*

## **13.4 Productization Requirements**

*Insert your content here.*

## **13.5 Release Requirements**

*Insert your content here.*

# **14 Maintainability and Support Requirements**

## **14.1 Maintenance Requirements**

*Insert your content here.*

## **14.2 Supportability Requirements**

*Insert your content here.*

## **14.3 Adaptability Requirements**

*Insert your content here.*

# **15 Security Requirements**

## **15.1 Access Requirements**

*Insert your content here.*

## **15.2 Integrity Requirements**

*Insert your content here.*

## **15.3 Privacy Requirements**

*Insert your content here.*

## **15.4 Audit Requirements**

*Insert your content here.*



## **15.5 Immunity Requirements**

*Insert your content here.*

## **16 Cultural Requirements**

### **16.1 Cultural Requirements**

*Insert your content here.*

## **17 Compliance Requirements**

### **17.1 Legal Requirements**

*Insert your content here.*

### **17.2 Standards Compliance Requirements**

*Insert your content here.*

## **18 Open Issues**

*Insert your content here.*

## **19 Off-the-Shelf Solutions**

### **19.1 Ready-Made Products**

*Insert your content here.*

### **19.2 Reusable Components**

*Insert your content here.*

### **19.3 Products That Can Be Copied**

*Insert your content here.*

## **20 New Problems**

### **20.1 Effects on the Current Environment**

*Insert your content here.*

### **20.2 Effects on the Installed Systems**

*Insert your content here.*

### **20.3 Potential User Problems**

*Insert your content here.*

### **20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product**

*Insert your content here.*

### **20.5 Follow-Up Problems**

*Insert your content here.*

## **21 Tasks**

### **21.1 Project Planning**

*Insert your content here.*

### **21.2 Planning of the Development Phases**

*Insert your content here.*

## **22 Migration to the New Product**

### **22.1 Requirements for Migration to the New Product**

*Insert your content here.*

### **22.2 Data That Has to be Modified or Translated for the New System**

*Insert your content here.*

## **23 Costs**

*Insert your content here.*

## **24 User Documentation and Training**

### **24.1 User Documentation Requirements**

*Insert your content here.*

### **24.2 Training Requirements**

*Insert your content here.*

## **25 Waiting Room**

*Insert your content here.*

## **26 Ideas for Solution**

*Insert your content here.*

## Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.
2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?