

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

Software Requirements Specification

for

Plagiarism Checker

Version 2.0

Prepared by: Grenel Sumabat

and

Dong Nguyen

2/19/2017

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

Table of Contents

Table of Contents.....1

Project Agreement and Statement of Work.....2

1. Background and Initial Obligations.....2

1.1 The Buyer.....2

1.2 The Seller.....2

1.3 The Project.....3

1.4 Other Legal Stuff.....4

2. Project Specification and Requirement.....4

2.1 Objectives and Background.....4

2.2 Operating Environment.....4

2.3 Designing.....5

2.4 Implementation Constraints.....5

3. System Features.....5

3.1 Variable Names.....5

3.1.1 Functional Requirements.....5

3.2 Comparing Code.....6

3.2.1 Functional Requirements.....6

3.3 Programming Languages.....6

3.3.1 Functional Requirements.....6

3.4 Syntax.....6

3.4.1 Functional Requirements.....6

3.5 Memory.....6

3.5.1 Functional Requirements.....6

3.6 Uploading Document.....7

3.6.1 Functional Requirements.....7

3.7 Feedback.....7

3.7.1 Functional Requirements.....7

3.8 Archive.....7

3.8.1 Functional Requirements.....7

61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

PROJECT SPECS AND REQUIREMENTS

Document version: 2/19/2017

Project Agreement and Statement of Work

THIS PROJECT AGREEMENT AND STATEMENT OF WORK (this “Agreement”) is entered into by and between the undersigned Buyer and Service Provider as of the Effective Date. The Buyer and Service Provider are sometimes referred to collectively herein as the “Parties” and individually as a “Party”. All capitalized terms not defined in this Agreement have the meanings given to such terms in the Terms of Service Agreement (“Terms of Service”) available, unless the context requires otherwise.

1. BACKGROUND AND INITIAL OBLIGATIONS

1.1 The “Buyer” is:

Frank Witmer
Computer Science & Engineering
ECB 308B
907-786-1637
Email: fwitmer@alaska.edu
Web: <http://www.cse.uaa.alaska.edu/~fwitmer/>

1.2 The “Seller” is:

Grenel Sumabat
510 Tyee Cir
Anchorage, Alaska 99503
Email: gsumabat2@alaska.edu

Dong Nguyen
518W 75th Ave
Anchorage, Alaska 99518
Email: dlnguyen@alaska.edu

1.3 The Project is:

106 The Plagiarism Checker is a program that validates a student's submitted
107 program.
108
109 Delivery Date:
110
111 1. Project Description
112
113 The Plagiarism Checker validates whether a student's submitted program
114 has been copied or plagiarized from other submitted programs. This
115 application will analyze and gather the similarities between the programs
116 that have been submitted to the database.
117
118 2. Key Assumptions
119
120 Uploading code to the checker field, then it will go through every single
121 line to check if the submitted code is copied or plagiarized from the other
122 sources.
123
124 3. Scope of Service
125
126 Comparing the given code to the public database. It will go through
127 some specific familiar webpages, such as StackoverFlow, Codein, etc.
128 This program will detect instances of plagiarism within the text document,
129 and will bring a tool to oppose the cut, copy and paste culture which is
130 strongly recommended at the core fundamental values of academic
131 integrity.
132
133
134 4. Milestone Deliverables
135

Milestone Deliverable (Task need to be done)	Delivery Date
1. Meeting with sponsor and get general idea about the project	February 2, 2017
2. Submission of first draft for requirements and specification	February 5, 2017
3. Submission of rough draft for visual aids	February 12, 2017
4. Submission of second draft for requirements and specification	February 19, 2017
5. Submission for final review	April 1, 2017
6. Final Delivery Date	April 15, 2017

136

137
138 **1.4 Other legal stuff:**
139
140 After final submission and testing of the application, the developers will hand
141 over the source code to the sponsor and will not have any responsibility of maintaining
142 the software.
143
144
145

146 **2. PROJECT SPECIFICATION AND REQUIREMENTS**

147
148
149 **2.1 Objectives and Background**

150
151 A “Plagiarism Checker” is a program building base on webpage interface. It helps
152 to check if student’s works are taken from any online resources without citation. The
153 program allows vast collections of documents to be compared to each other, making
154 successful detection much more likely.
155

156 Plagiarism checker is used for checking some courses at UAA such as CSCE A201, or
157 A202.

158 Database capacity will be able to hold all the data from required course.

159 The program will give a feedback about percentage of similarity between student’s code
160 and the online database with highlighting plagiarized lines.

161 Each file must be submitted with valid extension, unless the program will terminate with
162 an error message.
163

164 Our goal on this project is producing the final project meet sponsor expectation. The
165 “Plagiarism Checker” can help:

- 166
167 a) To detect instances of plagiarism to ensure honesty, trust, fairness, respect,
168 and responsibility.
169 b) To ensure academic integrity by detecting plagiarism.
170 c) To create the environment of academic morality and academic honesty.
171 d) To encourage academic sincerity and work originality among students.
172 e) To abase duplicity, cheating and fraudulent work.
173

174 **2.2 Operating Environment**

175
176 This program works on most common web browser such as Safari, Chrome, Firefox,
177 etc. For efficient and effective execution of the software, other requirements are:

- 178
179 • Database Server
180
181 • Global Database
182

- Internet Connection
- Search Engine

2.3 Designing

The program is put into system design, which can be used for future references. Designing focuses mainly on attributes such as:

- Data Structure
- Software Architecture
- Interface Representation
- Procedural Details

2.4 Implementation Constraints

The program will need internet connection because it is linked to a search engine. So, if there is problem with the internet connection then it could hamper the desired result. Search engine uses its own global database to find plagiarized code or code structure in the submitted works.

There are some constraints with this program:

- For checking of instances of plagiarism requires that the users should declare the specific programming language they want to check.
- Uploading of scanned images or screenshots is not allowed.
- Be sure that all syntax and braces are carefully checked before submitting.
- Should be used under trustful internet connection.

3. SYSTEM FEATURES

3.1 Variable Names

The application will verify if just the variable and class names were changed. This feature is a medium level priority.

3.1.1 Functional Requirements

226
227 After submission, if only the variable names and class names were changed, then they
228 will be highlighted to notify the user has plagiarized everything besides and just
229 changed the names.
230

231 **3.2 Comparing Code**
232

233 The similarities of submitted assignments will be detected by comparing parts of the
234 code with other submitted codes within the database. The feature is marked as a high-
235 level priority.
236

237 **3.2.1 Functional Requirements**
238

239 The similar codes should be highlighted to acknowledge which parts were too similar or
240 identical. If the compared codes don't have any similarities, then it has successfully
241 passed the checker.
242

243 **3.3 Programming Languages**
244

245 This application can validate various types of programming languages and not only
246 applicable to Java and C++. This feature is marked as a low-level priority.
247

248 **3.3.1 Functional Requirements**
249

250 TBD
251

252 **3.4 Syntax**
253

254 The application will check if the code only contains common syntaxes or similarities in
255 small simple code and if so, the user will be notified. This feature is marked as a low-
256 level priority.
257

258 **3.4.1 Functional Requirements**
259

260 TBD
261

262 **3.5 Memory**
263

264 This application will contain a back-end design to free up available memory space for
265 later use in the future. This feature is a medium level priority.
266

267 **3.5.1 Functional Requirements**
268

269 TBD
270

271 **3.6 Uploading Document**
272
273 This application will have a feature to upload a single document or multiple documents
274 to check whether they're in a readable format or not i.e. .zip, .cpp, .doc, .txt, or pdf. This
275 feature is marked as a high-level priority.
276
277
278 **3.6.1 Functional Requirements**
279
280 The user will be able to choose which file they'd like to import and upload to the web
281 application. If the document format is not applicable, then it will return an error message
282 saying that the format type is not supported with this application.
283
284 **3.7 Feedback**
285
286 This application will be able to send feedback to the user after submission of the
287 student's code, to be checked. This feature is marked as a medium level priority.
288
289 **3.7.1 Functional Requirements**
290
291 The feedback of the submitted code will be a percentage of how much of the submitted
292 program was detected as being plagiarized. If the given percent surpasses the minimum
293 requirement, then the student's code is valid. During batch processing, the program will
294 extract the student's name from code comments when uploaded.
295
296 **3.8 Archive**
297
298 This application will contain an archive of all submitted assignments within the recent
299 years. This feature is marked as a high-level priority.
300
301 **3.8.1 Functional Requirements**
302
303 The archive will have organized files from years past that are easily accessible. The
304 submitted programs will be checked against the archive database to ensure no
305 plagiarism is occurring with previous student's code.