|  |  |
| --- | --- |
| Code Similarity Comparer(CSC), User story 1 | |
| **Main Path** | **Alternate Paths** |
| 1. User submits a target source code file and an original source code file to the system  2. System format the target source code file and an original source code for comparer to compare more accurately  3. The system scans through two files line by line and count the number of identical lines based on identity of plain text until the end of shorter one. The system calculate the percentage of identical lines in all lines in target file  4. The system output the percentage to user |  |

|  |  |
| --- | --- |
| Code Similarity Comparer(CSC), User story 2 | |
| **Main Path** | **Alternate Paths** |
| 1. User submits a target source code file and an original source code file to the system  2. System format the target source code file and an original source code for comparer to compare more accurately  3. The system splits functions of two files and compares those functions by using longest common subsequence algorithm. The System calculates the similarity with the combination of function of highest similarity  4. The system output the percentage to user |  |

|  |  |
| --- | --- |
| Code Similarity Comparer(CSC), User story 3 | |
| **Main Path** | **Alternate Paths** |
| 1. User submits a target source code file and an original source code file to the system  2. System format the target source code file and an original source code for comparer to compare more accurately  3 The system extracts the variable names from two files and uses the extracted variable names to calculate the similarity using regular expression The System calculates the similarity by percentage of the number of variables in longest common subsequence  4. The system output the percentage to user | 3.1 The system extracts the keyword of C language in two vectors and calculate the similarity of two vectors using cosine similarity |

|  |
| --- |
| C:\Users\52710449\Desktop\cs3343\doc\ucd.PNG |

Use Case 1: Check similarity of 2 files

|  |  |
| --- | --- |
| Use Case Name | Check similarity of 2 files |
| Use Case Number | UC1 |
| Actor and stakeholder | User |
| Description | User check the similarity of 2 code files provided by user |
| Assumptions & Preconditions: | The code files to be compared are both using c language and able to compile ( no Syntax error) |
| Successful Guarantee: | The similarity result of two code files is calculated and displayed to user. |
| Main Successful Scenario | |
| Actor Actions | System Response |
| 1. User submits a target source code file and an original source code file to the system | 2. System format the target source code file and an original source code for comparer to compare more accurately  3.1 perform UC11  3.1 perform UC12  3.1 perform UC13  3.1 perform UC14  4. The system output the percentage or similarity to user |

|  |  |
| --- | --- |
| Use Case Name | Compare Line by Line |
| Use Case Number | UC11 |
| Actor and stakeholder | User |
| Description | User check the similarity of 2 code files using line by line method |
| Assumptions & Preconditions: | The code files to be compared are both using c language and able to compile ( no Syntax error) |
| Successful Guarantee: | The similarity result of two code files is calculated and displayed to user. |
| Main Successful Scenario | |
| Actor Actions | System Response |
|  | 1. Require UC 1  2. The system scans through two files line by line and count the number of identical lines based on identity of plain text until the end of shorter one. The system calculate the percentage of identical lines in all lines in target file |

|  |  |
| --- | --- |
| Use Case Name | Compare Function by function |
| Use Case Number | UC12 |
| Actor and stakeholder | User |
| Description | User check the similarity of 2 code files using function by function compare method |
| Assumptions & Preconditions: | The code files to be compared are both using c language and able to compile ( no Syntax error) |
| Successful Guarantee: | The similarity result of two code files is calculated and displayed to user. |
| Main Successful Scenario | |
| Actor Actions | System Response |
| 1. User submits a target source code file and an original source code file to the system | 2. System format the target source code file and an original source code for comparer to compare more accurately  3. The system splits functions of two files and compares those functions by using longest common subsequence algorithm. The System calculates the similarity with the combination of function of highest similarity.  4. The system output the percentage of similarity to user |

|  |  |
| --- | --- |
| Use Case Name | Compare by regex expression |
| Use Case Number | UC13 |
| Actor and stakeholder | User |
| Description | User check the similarity of 2 code files using regex expression method |
| Assumptions & Preconditions: | The code files to be compared are both using c language and able to compile ( no Syntax error) |
| Successful Guarantee: | The similarity result of two code files is calculated and displayed to user. |
| Main Successful Scenario | |
| Actor Actions | System Response |
| 1. User submits a target source code file and an original source code file to the system | 2. System format the target source code file and an original source code for comparer to compare more accurately  3. The system extracts the variable names from two files and uses the extracted variable names to calculate the similarity using regular expression The System calculates the similarity by percentage of the number of variables in longest common subsequence.  4. The system output the percentage of similarity to user |

|  |  |
| --- | --- |
| Use Case Name | Compare by cosine similarity |
| Use Case Number | UC14 |
| Actor and stakeholder | User |
| Description | User check the similarity of 2 code files using cosine similarity |
| Assumptions & Preconditions: | The code files to be compared are both using c language and able to compile ( no Syntax error) |
| Successful Guarantee: | The similarity result of two code files is calculated and displayed to user. |
| Main Successful Scenario | |
| Actor Actions | System Response |
| 1. User submits a target source code file and an original source code file to the system | 2. System format the target source code file and an original source code for comparer to compare more accurately  3. The system extracts the keyword of C language in two vectors and calculate the similarity of two vectors using cosine similarity  4. The system output the percentage of similarity to user |