DUPLICITY CHECKER

Team Name: BX12

Course Title: Software Project Lab 1

Course Code: SE 2112

Project Mentor

Falguni Roy
Assistant Professor
IIT, NSTU

Team Member

Armanur Rashid (ASH1925013M)
Sourav Debnath (ASH1925022M)
Sourav Barman (ASH1925030M)

Submission Date: 9 October, 2021

Table of Contents

Introduction ---------- 5

Software Project Description ---------- 5

Story ---------- 5

Requirements ---------- 5

Functional Requirement ---------- 5

Duplicity Check ---------- 6

Capitalization Check ---------- 6

Spell Check ---------- 6

Button Sound ---------- 6

Showing matching line ---------- 6

Save File ---------- 6

Highlight wrong word ---------- 6

Non-functional Requirement ---------- 7

Maintainability ---------- 7

Reliability ---------- 7

Performance ---------- 7

Space ---------- 7

Proposed Process Model ---------- 7

Types of model ---------- 7

Evolutionary Model ---------- 7

Project Team ---------- 8

Team Name ---------- 8

Members ---------- 8

Proposed Timeline and Actual Timeline ---------- 8

Tools ---------- 10

Language ---------- 10

IDE ---------- 10

Future Direction ---------- 10

Software Project Metrics ---------- 10

Code Level ---------- 10

LOC ---------- 10

NCLOC ---------- 10

CLOC ---------- 10

Average LOC in a Class ---------- 10

Density of Comments -- 10

Design Level -- 11

Package-- 11

Class --- 11

Package Name -- 11

Classes --- 11

Collaboration -- 12

Software Project Deliverables--- 13

Summary -- 13

Reference-- 13

Table of Figures

Figure 1 Evolutionary Process Model........8
Figure 2 Total Collaboration........12
Figure 3 Collaboration of Armanur Rashid........12
Figure 4 Collaboration of Sourav Debnath........12

Introduction

Our software project is Duplicity Checker. It’s a desktop application. Duplicity Checker is an application where the user can compare two or more files to see if there is any duplicity. We can also compare between two specific folders. Here folder checking will be done between the files of each folder. So our application will automatically detect the files of each folder. After comparing the folders or the files the percentage result of duplicity will be shown to the user. But if the compare would have done between two single file then the result percentage with the matching line between two files will be shown to user. Besides that, our application can also handle the capitalization problem. Our application will perfectly capitalize any docx file. As well as our application can also find and mark the wrong spelled word in a docx file. Finally, the application will be able to save the desired text in docx or pdf file.

Software Project Description

Story

As we know in educational institutions, when students are given an assignment or a home task then they copy the work of others. So we tried some prebuilt desktop applications. But none of them could compare between two folders. And some of them needed to buy the pro version of that application. For this reason, we took decision to make a completely free and offline version of duplicity checker application. So we have taken this project to identify those who copy and to stop plagiarism.

Requirements

The software requirements are description of features and functionalities of the target system. It is a condition or capability needed by a user to solve a problem or achieve an objective. Requirement convey the expectations of users from the software products. Requirements can be obvious or hidden, known or unknown, expected or unexpected from client’s point of view.

Software system requirements are often classified as functional and non-functional requirements.

Functional Requirement

These are the requirements that are the end user specifically demands as basic facilities that are system should offer [9]. The functional requirements of our projects are as follows: Duplicity Check, Capitalization Check, Spell Check, Button sound, showing matching line, Highlight wrong word, Save File etc.

Duplicity Check

- In the case of arbitrary selection, user can select the desired files by pressing ctrl button. It will check duplicity between those files. After clicking the check button, it will show the percentage matching with which file or show the matched line if matching is done between two files.

- In the case of folder selection, the user has to select folders and after clicking the check button, it will show the percentage matching with which file and also show the selected folder name with file quantity. It will show the matched line if matching is done between two files.

Capitalization Check

- After taking a file, it will check if the capitalization in the file is correct.
- If capitalization is not correct, it will be fixed after clicking on the check button

Spell Check

- It will search all the wrong words in a file with our stored words to see if there is any wrong word spelling.
- If wrong, it will highlight the wrong word.

Button Sound

- The home page has a button sound on and off option. The sound of the button can be turned on and off.

Showing matching line

- It will show all the result in descending order with file name. In case of single file comparison, the matching lines will also be shown.

Save File

- We can save the result of duplicity, also save the file after capitalizing or correcting the spell.

Highlight wrong word

- In spell check, after checking the file it will highlight the wrong words

Non-functional Requirement

These are basically the quality constraints that the system must satisfy according to the project contract [9].

Maintainability

The user can easily maintain the application.

Reliability

Our application will fulfill its assigned task in a given environment for input cases, assuming that the hardware and the input are free of error.

Performance

The performance of our application is very good. It gives very fast result.

Space

Our application size is decent compare to other existence application.

Proposed Process Model

A software process model is a simplified representation of a software process. Each model represents a process from a specific perspective.

Types of model

1. The Incremental Development Model.
2. The spiral model.
3. Evolutionary model.
4. The phases of iterative development.
5. The principles of agile methods.
6. The Waterfall Model

Evolutionary Model

Evolutionary model is a combination of iterative and incremental approach to software development. Evolutionary model is commonly used when the client wants to start using the core features instead of waiting for the full project. Evolutionary model is also used in object oriented software development because the system can be easily portioned into units in terms of object. The Evolutionary development model divides the development cycle into smaller, incremental waterfall models in which users are able to get access to the product at the end of each cycle. We have followed this model in our project because, in this model it reduces the error because the core modules get tested thoroughly. And a user gets a chance to experiment partially developed system.

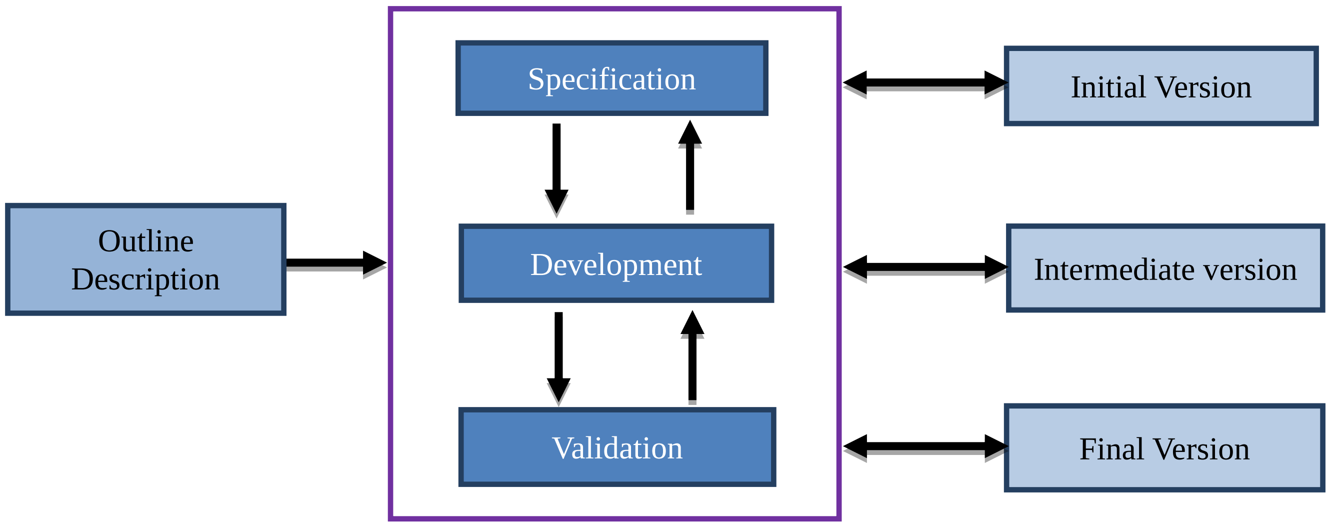

Figure 1 Evolutionary Process Model

Project Team

Team Name: BX12

Members

Armanur Rashid (ASH1925013M)
Sourav Debnath (ASH1925022M)
Sourav Barman (ASH1925030M)

Proposed Timeline and Actual Timeline

Proposed timeline is the timeline which are made by guess in a time bound and actual timeline means the work history which is make after completed work.

Table 01: Proposed Timeline Table

Task	**Deadline**
Proposal	Within 12th February 2020
Requirement Analysis, Specification	Within 1st March 2020
Designing, Study	Within 15th March 2020
Coding	Within 23th April 2020
Final Testing	Within 30th April 2020

****** Educational institutions were closed due to the **Corona epidemic**. So it was not possible to follow the time mentioned in the proposed timeline.

Table 02: Actual Timeline Table

Task	Deadline
Proposal	Within 13th February 2020
Requirement Analysis, Specification	Within 1st March 2020
Designing, Study	Within 23th April 2020
Coding	Within 27th July 2020
Final Testing	Within 2nd October 2021

Requirements Traceability Matrix

Traceability matrix or software testing traceability matrix is a document that traces and maps the relationship between two baseline documents that require a many-to-many relationship to check the completeness of the relationship [8].

Req1 = Duplicity Check (Single File)

Req2 = Duplicity Check (Multiple File)

Req3 = Duplicity Check (Folder)

Req4 = Show Matching line

Req5 = Capitalization Check

Req6 = Spell Check

Req7 = Highlight wrong word

Req8 = Button Sound

Req9 = Save File

Table 03: Requirement Traceability Table

Requirement Test Case	Req1	Req2	Req3	Req4	Req5	Req6	Req7	Req8	Req9
Test Case 1								✓	
Test Case 2	✓			✓					✓
Test Case 3		✓		✓					✓
Test Case 4	✓	✓	✓	✓					✓
Test Case 5	✓	✓	✓	✓					✓
Test Case 6					✓				✓
Test Case 7						✓	✓		✓
Test Case 8						✓	✓		✓
Test Case 9	✓	✓	✓	✓	✓	✓	✓		✓

Tools

Language: JAVA

IDE: An IDE (Integrated Development Environment) contains a code editor, a compiler or interpreter, and a debugger, accessed through a single graphical user interchange (GUI). Our IDE is Intellij IDEA.

Future Direction

We will work on whether we have copied any text from any online site. We will do more with spell checker. We will work with word suggestion and replace correct word with wrong word.

Software Project Metrics

Code Level

LOC - Line of code in the whole project. The Total number of code lines in whole project.

NCLOC - Non-comment line of code. The line which are not commented and this code accomplish our objective.

CLOC - Comment line of code. The line of code which are not working to calculate output.

Average LOC in a Class - The average number of line of code in each class.

Density of Comments - It means the number of comment line proportion to average code.

Table 04: Code Level Table

LOC	2376
NCLOC	2253
CLOC	123
Average LOC	125
Density Of Comments	5.2%

Design Level

Package: Package in Java is a mechanism to encapsulate a group of classes, sub packages and interfaces.

Class: A class, in the context of Java are templates that are used to create objects, and to define object data types and methods.

Package Name: Duplicity_Checker_Project

Classes

1. About_us
2. Back_Button
3. Basic_Frame_Duplicity
4. ButtonSound
5. Capitalization
6. Check_extension
7. Duplicity_Folder
8. Duplicity_Random_File
9. File_read
10. Frame_Container_Template
11. Highlight_text
12. Home
13. MainClass
14. Panel_BackButton_Template
15. Save_file
16. Sorting_Result
17. Spell_Check
18. Splash_Screen
19. User_Guidelines

- Static Variable Percentage: 1.012%
- Methods Per class (Average): 2

Collaboration

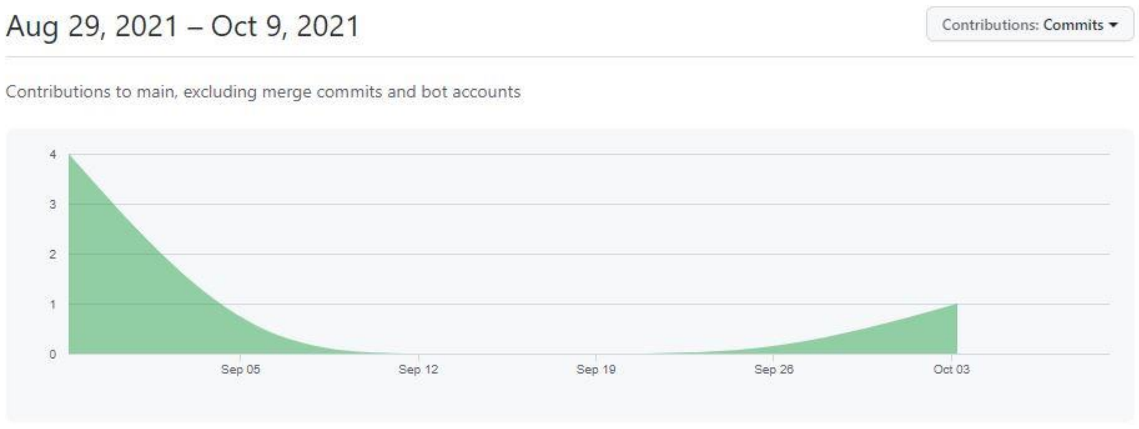

Figure 2 Total Collaboration

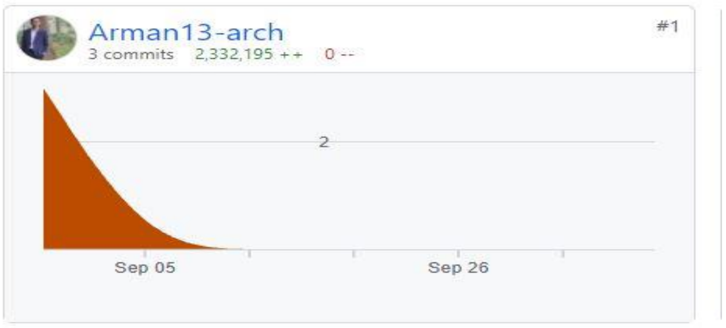

Figure 3 Collaboration of Armanur Rashid

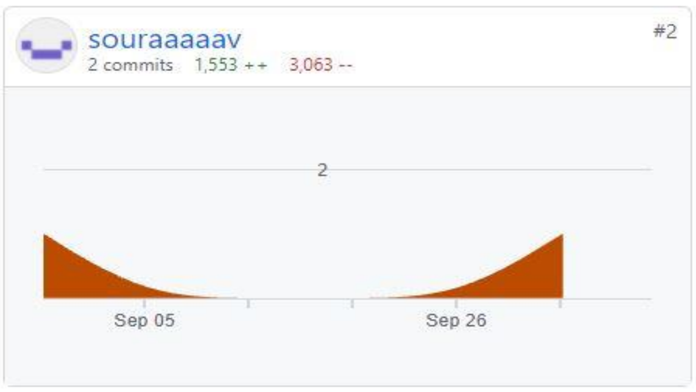

Figure 4 Collaboration of Sourav Debnath

Software Project Deliverables

1. PowerPoint Microsoft Open XML
2. Project demo video
3. Project report
4. Softcopy user manual
5. Source Code

Summary

As we can see copying in an assignment is as easy as drinking water. It will hamper student in their future. But using our application we can easily find out the duplicity percentage. So this application will stop the copying in assignment, exam or similar activities. Our application will also help to properly capitalize the wrong case text. Another thing is that our application will also help to find and mark the wrong spelled word. Finally, a user can save the desired text in docx or pdf format. Besides that, this project helped us increasing our knowledge for Object Oriented Language (JAVA) and getting experience in java GUI.

Reference

[1] Herbert Schildt (2018). Java the Complete Reference (10th edition). McGraw-Hill Education.

[2] Ian Sommerville (2011). Introduction to Software Engineering (9th edition). Pearson.

[3] https://mkyong.com/java/java-read-and-write-microsoft-word-with-apache-poi/ [Accessed at 22.6.2020]

[4] https://stackoverflow.com/questions/16682942/reading-docx-file-in-java [Accessed at 23.6.2020]

[5] https://orangefreesounds.com/music/background-music/page/2/ [Accessed at 12.8.2020]

[6] https://en.wikipedia.org/wiki/Code_smell [Accessed at 01.10.2021]

[7] https://en.wikipedia.org/wiki/Code_refactoring [Accessed at 01.10.2021]

[8] https://www.testbytes.net/blog/software-testing-traceability-matrix/ [Accessed at 06.10.2021]

[9] https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/ [Accessed at 06.10.2021]