Algorithms for the Generation of an Electronic Customized Answer Booklet for Universities

Doreck Nduhukire[1], Evarist Nabaasa[1], Deborah Natumanya[1]

[1]Mbarara University of Science and Technology, Uganda

Corresponding Author: Deborah Natumanya; Email: deborahnatumanya@must.ac.ug

A B S T R A C T

Academic institutions in Uganda are accustomed to using pen-on-paper exams where students are given an answer booklet at the time of the exam. Universities frequently use customized answer booklets which help to lower the prevalence of exam fraud among students. Universities continue to encounter the challenge of examining summative questions like essays after the migration to electronic examinations brought about by the Covid-19 social distancing effects due to the lack of a customized electronic answer booklet. The primary goal of this research was to design algorithms for a customized electronic answer booklet. These algorithms went through six stages, starting with analyzing the problem and ending with developing the algorithms. The design, analysis, and coding of four algorithms i.e. cover page, single empty page, adding a page, and watermark addition algorithms was completed. The designed electronic personalized answer booklet is dynamic in nature and supports page addition, text crossing out or rubbing, and navigating through the electronic question paper. Any academic institution may utilize this electronic answer book.

INTRODUCTION

During the three-year Covid-19 epidemic phase, academic institutions, particularly universities, were forced to make a major transformation from traditional pen-on-paper exams to electronic exams (Gudel, 2022; Revilla-Cuesta, Skaf, Varona, & Ortega-López, 2021). Traditional pen on paper is the act of examinations where students are presented with hard copy question papers and answer booklets where to write their corresponding answers (Guimarães et al., 2018; Sindre & Vegendla, 2015) while electronic examinations refer to the act of using electronic devices to take on an examination which may or may not be limited by location (Bashitialshaaer, Alhendawi, & Lassoued, 2021; Rajala et al., 2016). Electronic learning was quickly adopted by schools to ensure continuity of instruction in the hopes that the situation that had caused school closure, distance learning, and the university to resume normal operations would calm down (Karakose, 2021; Tadesse & Muluye, 2020). Unfortunately, it lasted longer, necessitating the need for electronic assessment to conduct progressive assessments that included course work, assignments, and exams (Kahunde, Ogwang, Maya, Paul Lakuma, & Sserunjogi, 2022; Ssenkusu, Ssempala, & Mitana, 2022).

As they offered a platform for creating brief tests or quizzes, the electronic learning platforms that were being used to conduct electronic learning sessions could easily allow progressive assessment (García-Alberti, Suárez, Chiyón, & Mosquera Feijoo, 2021; Singh, Steele, & Singh, 2021). However, providing summative or end-of-semester exams presented a dilemma, and most universities were forced to cease examinations as a result of the absence of an appropriate method or platform (Guangul, Suhail, Khalit, & Khidhir, 2020; Noorbehbahani, Mohammadi, & Aminazadeh, 2022). More so, the university answer booklet's standards were not met by the text editors or writing programs like Microsoft Word that were readily available. In light of this, the goal of this project

was to design algorithms for a customized electronic answer booklet that supports free electronic handwriting.

Universities frequently used hardcopy answer books, and the proposed algorithms were to electronically convert the hardcopy answer book while keeping all of the answer book's rules. The answer booklet is divided into various sections, including the face page, blank pages, header, and footer. Each of these sections also includes a variety of components: the face page has tables that hold various vital details about the student, including their name, program, course name, course code, exam date, etc.; exam score information, including the marks given for each question and the overall score; instructions, which explain how to use the

booklet; a header (the student's number); and a footer (booklet serial number and list of questions attempted).

METHODS

An algorithm, according to (Levitin, 2008), is described as a series of clear instructions for resolving a problem, or for attaining the desired result for any valid input in a predetermined amount of time. Algorithms are viewed as methodical approaches to issues. These solutions are not the answers; rather, they are detailed directions for finding the answers. A typical process for designing and assessing algorithms is shown in Figure 1 as a series of steps.
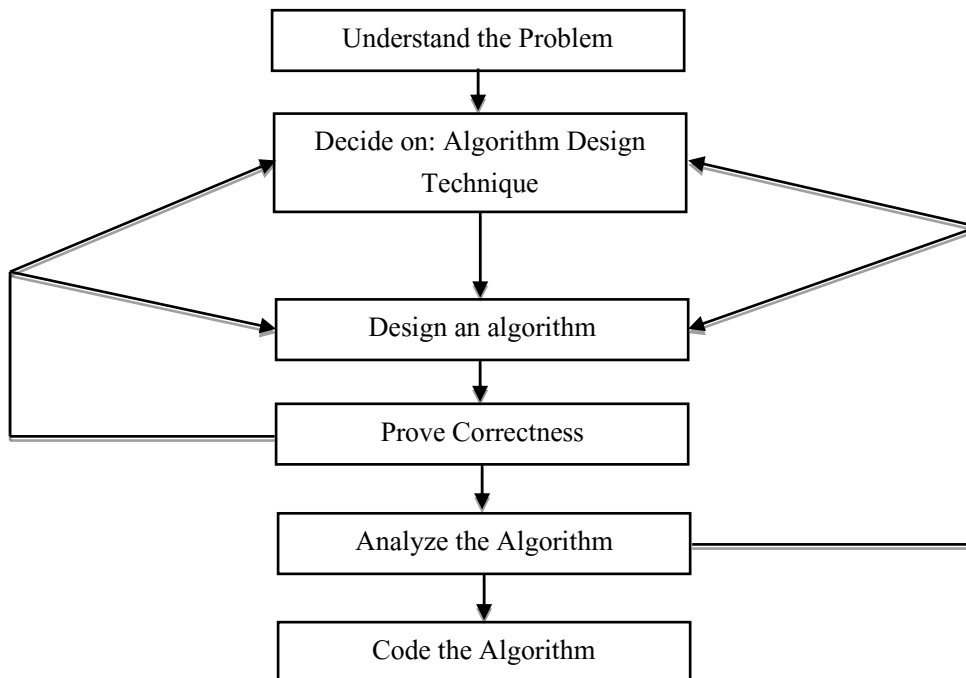

Figure 1. Steps Followed in the Design of Algorithms

All of the information on each page of the hard copy answer booklet was noted down for consideration after rigorous analysis. The information on each page was eventually organized into groups based on the various functions it performs to create a reasonable workload from which an algorithm could be developed. For the electronic answer booklet, the following algorithms had to be taken into account: cover page, empty page, adding an empty page, and watermark addition.

Given the nature of the problem that needed to be solved, the divide-and-conquer design technique was used to guide the creation of these algorithms

(Osa et al., 2018). The divide and conquer technique uses a top-down strategy that involves splitting the large problem into smaller ones and then solving each one on its own, recursively. After then, the solutions are blended to find one that addresses the original issue. The single empty page algorithm, cover page algorithm, method for adding an empty page, and algorithm for watermark insertion were among the algorithms designed.

The steps of each algorithm were afterward documented and confirmed or examined for accuracy by two professionals in the field of algorithm design, one of whom was a computer scientist and the other a mathematician. The

algorithms' worst-case complexity was also evaluated as part of the analysis. The algorithms were later developed to create an electronic exam answer booklet.

In this study, an algorithm's complexity is defined as a measurement of how long it takes to run given an input of size n. The complexity measured in this study focuses less on how long an algorithm really takes to run and more on how much labor it involves. To find areas where the algorithms needed to be improved, complexity was measured.

Based on the worst-case scenario, a theoretical analysis of time efficiency was conducted. In worst-case scenario analysis, the execution time of an algorithm is estimated at its maximum bound when it is fully supplied with inputs. The complexity of each method was calculated by calculating the number of basic operations repeated in relation to the size of the input. The operation that was deemed to be fundamental was the one that made the biggest difference in how long the algorithm took to run. To better understand how well the planned algorithms performed, a worst-case complexity analysis was conducted. This provided a foundation for future efficiency improvements. The equation below was used to calculate the worst-case complexity of each of the algorithms.

$$W_{(n)} = maxt\ (I)\ I \in D_n$$

Where:

D_n was the set of inputs of size n for the problem under consideration,

I was an element of D_n.

$t(I)$ was the number of basic operations performed by the algorithm on input I.

The function W is called the worst-case complexity of the algorithm. $W_{(n)}$ is the maximum number of basic operations performed by the algorithm on any input of size n.

Following the above function, a basic operation for each algorithm was identified and complexity $W_{(n)}$ for each algorithm was calculated.

RESULTS AND DISCUSSION

The divide and conquer technique is a top-down strategy in which an issue is broken down into smaller sub-problems that are solved on their own to arrive at a solution for the larger problem. As seen in Figure 2 below, the divide and conquer strategy was used to successfully complete the electronic answer booklet task.
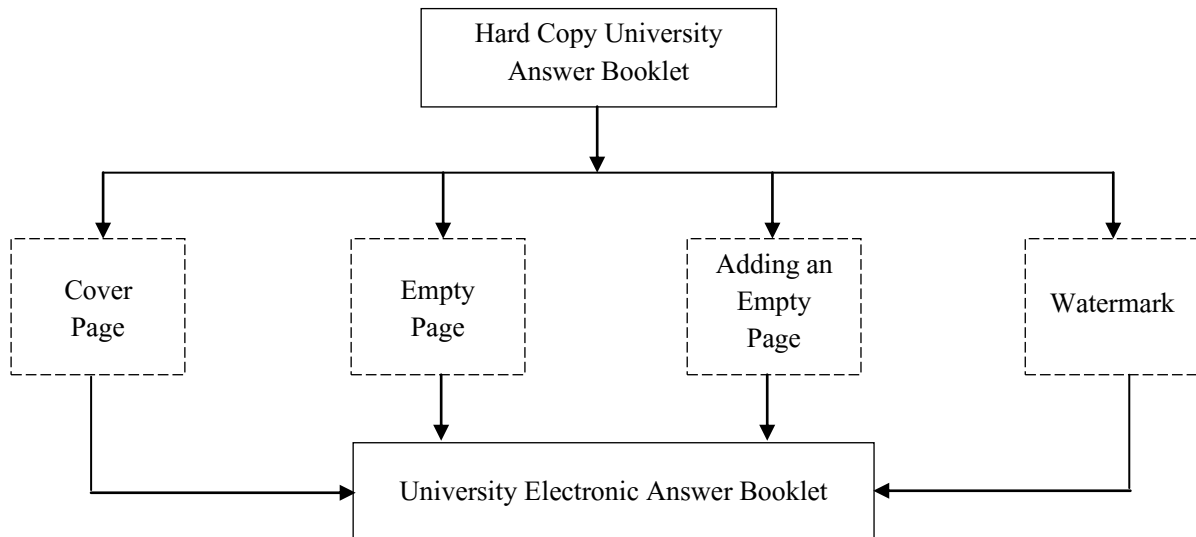

Figure 2. Electronizing the Electronic Examination Answer Booklet

Cover Page Algorithm

Input: TH = header, TF=footer, T=table, C=Cells, Col= Columns, Set P $\{p_1, p_2, p_3, …, p_n\}$ = paragraphs

Output: P_N customized cover page of the Answer Script booklet:

1: Open directory F and edit its empty page *Pn*

2: N ← *i*

3: Create a Tuple Y = (TH, TF, T, C, Col, P)

4: Initialize Elements of Y

5: Create a Table TH = (th, Colh, Celh, Caph)

6: Initialize Elements of TH

7: Create a Table TF = (P, Tf, Colft, CF)

8: Initialize Elements of TF

9: Create a Table T = (P1, P2, P3, T1, Col1, Cap1, Cel 1, L)

10: Initialize Elements of T

11: Add TF, FT, and T to *Pn*

12: Update *Pn* and Stop

This algorithm creates the cover or face page of the examination answer booklet. The face page contents like the students' details, examiners remarks, footer, and header are added by this algorithm by the use of tables. Worst Case Analysis; W(n)=6n because there are 6 maximum basic operations at lines 1, 3, 5, 7, 9, and 11.

Single Empty Page Algorithm

Input: N = number of pages, F = file directory

Output: (Single empty page)

1: Create a Tuple T (N, F, i)

2: **if** F doesn't exist **then**

3:        Create directory F

4:        *N* ← *I* and i = 0

5: **else**

6:        Overwrite existing directory

7: N = i

8: Create *Pn*

9: Save *Pn* into F

10: Exit

This algorithm generates an empty page of an answer booklet and stores the empty page in a file directory. From the Single Empty Page Algorithm, Worst Case Complexity: W(n)=n, this is because the algorithm executes the basic operation only once at line 8

Adding a Page

Input: f_i = pages number; F= directory; nxtp = next page

Output: F with page P_n

1: N ← (*i* + 1)

2: *nxtp* ← *False*

3: **if** N > 0 **then**

4: Initialize nxtp = True

5: **if** nxtp **then**

6:        insert Nth page

7:        Create table Th=(celh, capth, colh)

8:        Initialize elements of Th

9:        Create table Tb=(celb, colb)

10:        Initialize Elements of Tb

11:        Create table Tf=(celf, colf captf)

12:        Initialize elements of Tf

13:        Add Th, Tb, and Tf to *Pi* + 1

        end if

    end if

14: End

A student can add a page in the answer booklet with the help of this algorithm. This is done by open and empty page and giving it properties like the header, body, and footer. Its maximum number of basic operations are 4 at lines 4,7,9 and 11. Therefore, Worst Case Complexity: $W(n)=4n$.

Watermark Addition Algorithm

Input: w = Watermark, f = file directory, i = page number

Output: File P_n with customized answer booklet page with a watermark logo.

1: Open f and edit its page *Pn*

2: $n \leftarrow (i + 1)$

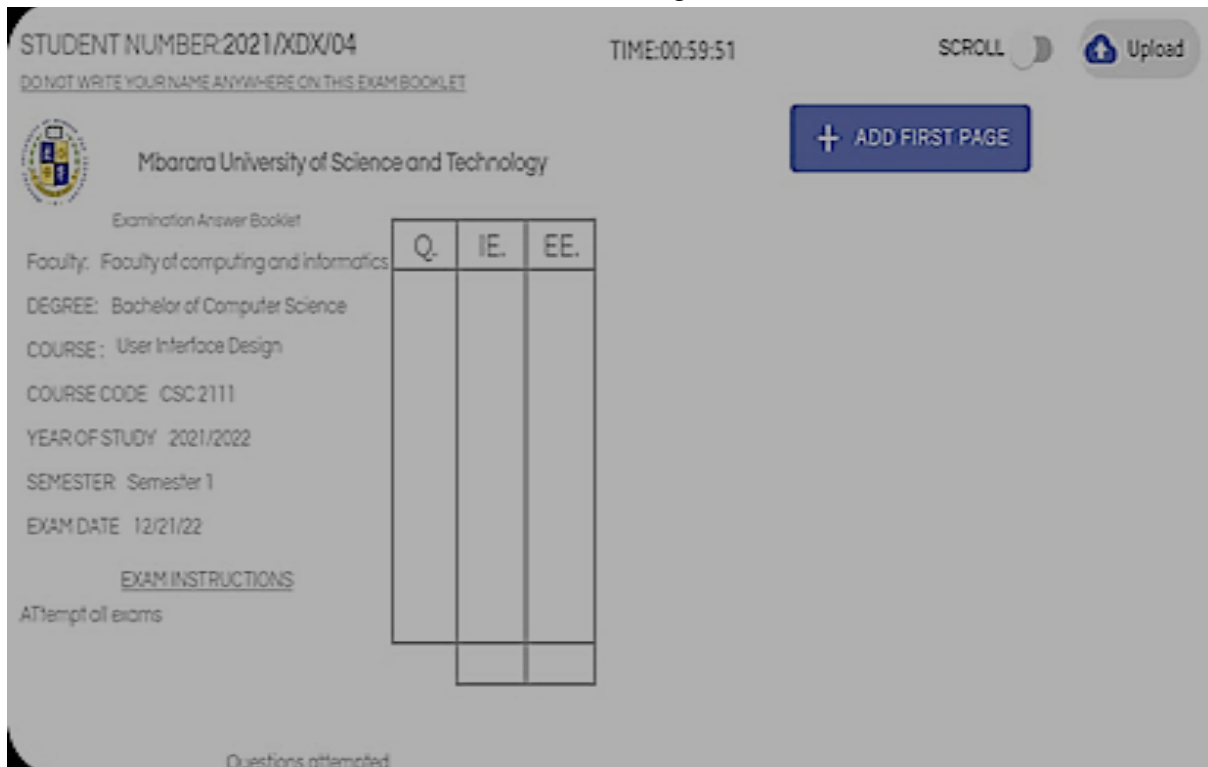3: Initialize w to watermark.png

4: if *Pi* <=*Pn* then

5:      Add w to page *Pn*

6: Update *Pn* and STOP

This algorithm adds a watermark within the examination answer booklet. For this to happen, an answer booklet file is opened and a watermark which is the university logo is added as an image. Worst Case Complexity; $W(n)=n$, this is because the algorithm executes the basic operation only once at line 5.

Algorithm Coding

The algorithms were programmed to produce a comprehensive, university-specific electronic answer booklet. The answer booklet was created to work in a dynamic setting where the booklet's page count might be expanded using a button-activated new page addition method. In case a mistake was made, the booklet also allows the user/student to wipe out material and cross out pages. The student can effortlessly flip between the question paper and answer booklet at the same time. Below are sample images of the customized electronic answer booklet.
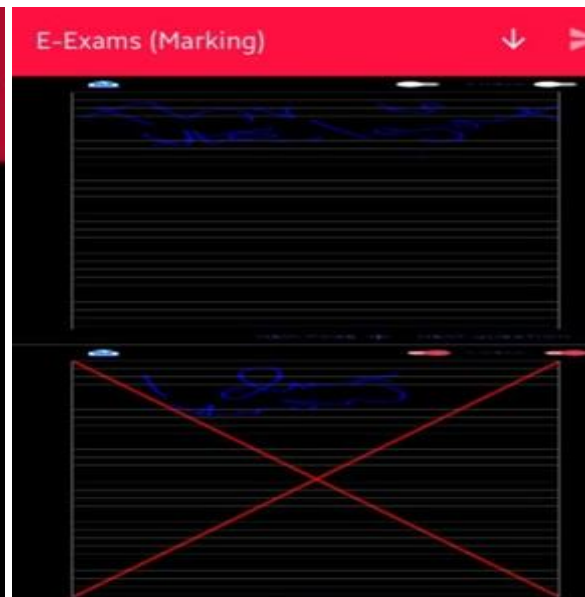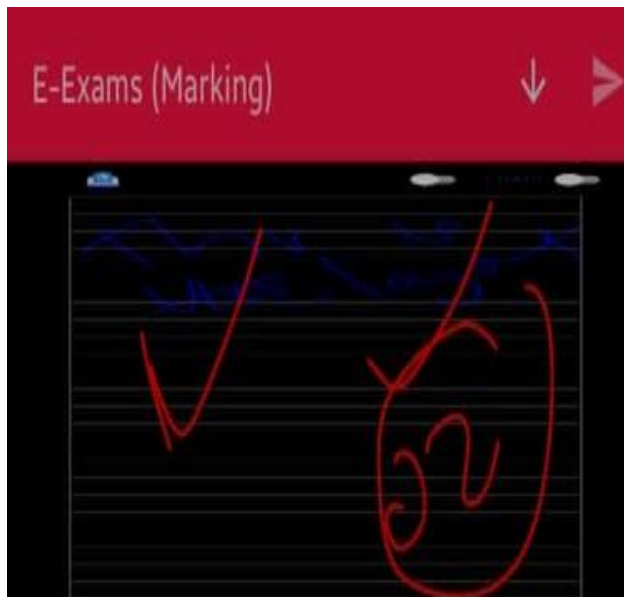
The designed algorithms underwent validation, coding, and efficacy testing. The worst-case scenario for each algorithm was measured to serve as a foundation for future enhancements to the algorithms, enhancing their effectiveness and efficiency. This was done using the most fundamental operations that an algorithm could possibly perform in the worst-case situation, such as adding an empty page to an answer booklet. We determined the upper bound on each algorithm's execution time for the worst-case complexity study. We needed to understand the scenario in which most operations are carried out. The worst case was estimated for each algorithm and is shown in the results section above.

The electronic answer booklet was created with the following features, including a face page that includes the university logo, university name,

examination instructions, student information, a section for internal examiners' scored marks and external examiners' comments, booklet serial number, and academic year. As soon as a student logs into the examination system, the system instantly fills out the face page information. The following pages of the answer booklet that the student can edit include a text area where they can write all of their answers, a header with buttons for erasing and canceling, a text field for each attempted question, and a footer with buttons for the next question, previous question, and add a new page.

A student can use the electronic free handwriting function to write in the electronic answer book, use an eraser, and even cancel an entire page if they make a mistake. The margins on the left and right of the student's answer booklet are restricted and cannot be written in; however, the lecturer may alter these margins.

CONCLUSION

This study sought to electronicize the regularly used hard copy examination answer booklet that would be utilized in the framework designed for electronic examinations (Natumanya & Nabaasa, 2022). The algorithms are prepared to be implemented in the framework because they have been fully developed and analyzed as evidenced by the findings reported above. The algorithms were developed, and the answer booklet output is highly interactive because students may write in it, mark it with a pen, cross things out, and add blank pages, while lecturers can grade the answer script without changing the students' responses.

REFERENCES

1. Bashitialshaaer, R., Alhendawi, M., & Lassoued, Z. (2021). Obstacle comparisons to achieving distance learning and applying electronic exams during Covid-19 pandemic. *Symmetry, 13*(1), 99.
2. García-Alberti, M., Suárez, F., Chiyón, I., & Mosquera Feijoo, J. C. (2021). Challenges and experiences of online evaluation in courses of civil engineering during the lockdown learning due to the Covid-19 pandemic. *Education Sciences, 11*(2), 59.
3. Guangul, F. M., Suhail, A. H., Khalit, M. I., & Khidhir, B. A. (2020). Challenges of remote assessment in higher education in the context of Covid-19: a case study of Middle East College. *Educational assessment, evaluation and accountability, 32*(4), 519-535.
4. Gudel, H. (2022). After a 2-year lockdown, schools in Uganda have reopened–but for many it may be too late. *LSE Covid-19 Blog*.
5. Guimarães, B., Ribeiro, J., Cruz, B., Ferreira, A., Alves, H., Cruz-Correia, R., . . . Ferreira, M. A. (2018). Performance equivalency between computer-based and traditional pen-and-paper assessment: A case study in clinical anatomy. *Anatomical sciences education*, *11*(2), 124-136.
6. Kahunde, R., Ogwang, A., Maya, D., Paul Lakuma, C., & Sserunjogi, B. (2022). School Reopening in Uganda After the Covid-19 Lockdowns: Preparedness, Attrition, Challenges and Prospects for Recovery.
7. Karakose, T. (2021). The impact of the Covid-19 epidemic on higher education: Opportunities and implications for policy and practice. *Educational Process: International Journal, 10*(1), 7-12.
8. Levitin, A. (2008). *Introduction to Design and Analysis of Algorithms, 2/E*: Pearson Education India.
9. Natumanya, D., & Nabaasa, E. (2022). An Electronic Examinations Framework with Electronic Free Handwriting. *Indonesian Journal of Innovation and Applied Sciences (IJIAS), 2*(2), 88-92.
10. Noorbehbahani, F., Mohammadi, A., & Aminazadeh, M. (2022). A systematic review of research on cheating in online exams from 2010 to 2021. *Education and Information Technologies*, 1-48.
11. Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., & Peters, J. (2018). An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics, 7*(1-2), 1-179.
12. Rajala, T., Kaila, E., Lindén, R., Kurvinen, E., Lokkila, E., Laakso, M.-J., & Salakoski, T. (2016). *Automatically assessed electronic exams in programming courses*. Paper presented at the Proceedings of the Australasian computer science week multiconference.

13. Revilla-Cuesta, V., Skaf, M., Varona, J. M., & Ortega-López, V. (2021). The outbreak of the Covid-19 pandemic and its social impact on education: Were engineering teachers ready to teach online? *International journal of environmental research and public health, 18*(4), 2127.

14. Sindre, G., & Vegendla, A. (2015). *E-exams versus paper exams: A comparative analysis of cheating-related security threats and countermeasures.* Paper presented at the Norwegian Information Security Conference (NISK).

15. Singh, J., Steele, K., & Singh, L. (2021). Combining the best of online and face-to-face learning: Hybrid and blended learning approach for Covid-19, post vaccine, & post-pandemic world. *Journal of Educational Technology Systems, 50*(2), 140-171.

16. Ssenkusu, P. M., Ssempala, C., & Mitana, J. M. V. (2022). The Impact of Covid-19 on Schools: A Critical Analysis. *American Journal of Educational Research, 10*(4), 253-256.

17. Tadesse, S., & Muluye, W. (2020). The impact of Covid-19 pandemic on education system in developing countries: a review. *Open Journal of Social Sciences, 8*(10), 159-170.