

**Improve Learning and Teaching Activities by Incorporating
Version Controlling Concepts to Learning Management
Systems**

Amarasinghe N.G.C.M.

158750H

Faculty of Information Technology

University of Moratuwa

June 2018

**Improve Learning and Teaching Activities by Incorporating
Version Controlling Concepts to Learning Management
Systems**

Amarasinghe N.G.C.M

158750H

Dissertation submitted to the Faculty of Information Technology, University
of Moratuwa, Sri Lanka for the fulfilment of the requirements of Degree of
Master of Science in Information Technology

June 2018

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student

Signature of Student

Amarasinghe N.G.C.M.

.....

Date

Supervised by

Name of Supervisor

Signature of Supervisor

Chaman Wijesiriwardana

.....

Date

Dedication

We dedicate the output of this research work and thesis to development team of existing Learning Management Systems. Also I specially dedicate this new features to all those who generously contributed their valuable time, advising and helping in doing this research, especially to my supervisor Mr. Chaman Wijesiriwardana. And I hope the research and the findings described below will provide a useful insight for applying version controlling mechanisms to existing Learning Management Systems.

Acknowledgement

First of all, I would Like to thank my research supervisor Mr. Chaman Wijesiriwardana who spent his valuable time in guiding this research to make it a success. Furthermore, my next big thank goes to Prof. Asoka Karunanandha who taught us Research Methodology and Literature Review and Thesis Writing subjects which were the basis for this research.

Not only that my thanks should go to all the lecturers in M.Sc in Information Technology degree program of Faculty of IT, who gave their hands to sharpen our knowledge and ideas throughout these two years as they were the illumination which lit up our pathways to success.

Apart from the people who were directly involved, many more helped to make this project successful. Especially the people who engage with two surveys I have done with this research at the beginning and at the end as the evaluation part. Finally, I would like to thank all the batch mates of the M.Sc in IT degree program who gave their valuable feedback to improve the results of the research.

Abstract

E-Learning is becoming a most needful part of our education system. To the combination of the teacher and the student is more important like teaching something. As one of the major features of the existing e learning systems, “uploading documents” have become to a major functionality. Within this research, the researcher focuses to do needful validations to this major upload functionality (When updating the uploads previously done by the same user). After doing some initiate survey among the stakeholders of LMSs, identified the requirements exactly what the researcher has to do. Also as a software developer, the researcher always took the experiences from the version controlling systems (software developers use) like Bitbucket and GIT, which are in his working environment.

For full fill the requirements, the researcher found sources of some existing LMSs (Media Share android 1.1 and Moodle 3.5) and tried to apply the solutions on that. As a critical step, take a time to compare two existing systems like GIT and moodle (a version controlling system and an LMS), summarized the features, nice to have in LMSs while uploading. Then could generate own mechanism to compare two files and give a good validation for the files while uploading contents to the same place. And the researcher could process a SUS base survey among the lecturers and students about the new functions and took the SUS final result for that. And as the final result, the survey gives "Ok, we can improve more". And could handle a performance test for whole new features and could prove the performance are an acceptable level after the new implementation.

The reason behind the result is the researcher could give a good outcome from the research which could keep the existing patterns of the existing system and the new methods could stay with existing interfaces and keep the other functions as it is. Users of LMS got the new features without not taking any difficulties. And could give an answer to an existing problem which was in uploading files. This research didn't focus on PDF file uploading, as a future task to do, the researcher like to specially mention that. And this is only focused on texts. In future, researchers who find solutions for content comparison and uploading can think about the files which contain videos, audios, and images also.

Table of Contents

Abstract	iv
Chapter 1	1
Introduction	1
1.1 Introduction	1
1.2 Aim and Objectives	3
1.3 Background and Motivation	3
1.4 Problem in Brief	4
1.5 Proposed Solution.....	5
1.6 Structure of the Thesis	6
1.7 Identified Limitations	6
1.8 Summary.....	7
Chapter 2	8
Review of others works	8
2.1 Introduction	8
2.2 Use of content management in various systems.....	8
2.3 Summary of Challenges	10
2.4 Problem Definition	10
2.5 Summary.....	10
Chapter 3	11
Technology	11
3.1 Introduction	11
3.2 Technology basics	11
3.3 Pseudocode for the content comparison in file upload validation.....	12
3.4 Summary.....	13
Chapter 4	14
Approach	14
4.1 Introduction	14
4.2 Hypothesis	14
4.3 Input.....	14
4.4 Output.....	15
4.4.1 UI Changes	15
4.4.2 Performance improvement	15

4.5 Users	16
4.6 Summary.....	16
Chapter 5	17
Design and Implementation	17
5.1 Introduction	17
5.2 Implementation overview	17
5.1.1 Implement the functionalities	17
5.2 Implement the Features.	21
5.2.1 Track the changes of strings in lecturer’s panel.	21
5.2.2 Media Share android integrations involve with existing UIs	23
5.2.3 API for the data retrieving	25
5.2.4 File validates when student uploading multiple files at once.....	25
5.3 Multiple file uploading feature - Integrate of “Moodle 3.5” web application.....	26
5.4 Summary.....	28
Chapter 7	29
Evaluation.....	29
7.1 Introduction	29
7.2 Evaluation.....	29
7.2.1 Performance of the new features	29
7.2.2 Usability Scaling	30
7.2.4 Summary.....	33
Chapter 8	34
Conclusion and Future Work	34
8.1 Introduction	34
8.2 Conclusion.....	34
8.3 Future Work.....	35
8.4 Summary.....	35
Chapter 9	Error! Bookmark not defined.
References	
Appendix A - Source code to “Media Share 1.1 android”	38
Appendix B - Integrated to “Moodle3.5”	43
Appendix C calculating the speed performance of the MD5 and SHAs.	46
Appendix D – SUS Calculation	43

Table of Contents for Figures

figure 5.1 – Selecting a course and an assignment. (this is from existing application and keep them as it is).	22
Figure 5.2 – suggested features in assignment creation area. When uploading new file, system gives alert, informing that	23
Figure 5.2 – suggested features in assignment creation area. When uploading new file, system gives alert, informing that.	24
Figure 5.3 – suggested features in assignment creation area. When change the due date of the assignment, system alerting.	24
Figure 5.4 – suggested features in assignment creation area. When uploading a same file again, system gives an alert, informing that.	24
Figure 5.5 – suggested features in assignment creation area. When uploading a new file replacing the existing one, system gives an alert, informing that.....	24
Figure 5.6 – suggested features in assignment creation area. Emailing the changes to the users/students.	25
Figure 5.7– suggested features in assignment creation area. Users can see the history of the file uploaded.	25
Figure 5.8 – The existing file uploading are in the Moodle 3.5 web application...25	25
Figure 5.9 – Users can see what are the files he has already uploaded.....	26
Figure 5.10 – Suggested file uploading area in the app.	27
Figure 5.11 – When user uploading new files, system take them without no doubts.	27
Figure 5.12 – If the user upload same file again, system detect that and not upload, reject the uploading, select only new files and inform it to the users.	27
Figure 5.12 – If the user uploads the same file again, system detects that and not upload, reject the uploading, select only new files and inform it to the users.	28

Table of Contents for tables

Table 1.1 comparison the requirements with current Bitbucket	05
Table 7.1 evaluate performance for existing system	29
Table 7.2 evaluate performance for existing system	30
Table 7.3 evaluate performance for Introducing system	32
Table appendix 1 hashing performance part 1	50
Table appendix 2 hashing performance part 2	50
Table appendix 3 hashing performance part 3	50
Table appendix 4 hashing performance part 4	51
Table appendix 5 hashing performance part 5.....	51
Table appendix 6 hashing performance part 6	51
Table appendix 7 hashing performance part 7	52
Table appendix 8 appendix 8 evaluate SUS calculation for Introducing system	55