Secured Submission and Plagiarism Detection for a Web Based Computer Assisted Assignment Checking System

J. Mahata, *Member, IACSIT,* A. Mahata, *Member, IACSIT & IEDRC,* and S. Roy

***Abstract*—This paper intends to make one of the most complex, time-consuming tasks lab instructor face as simple as a few keystrokes. We propose a software to automate the assignment checking system which can be performed online in safe, secure and confidential manner. Lot of works has been tried in this field but we uniquely identified six categories of all programming assignments. Any programming language which will run as executable can be tested using this system. Almost all undergraduate and postgraduate courses belongs to science, technology or vocational education contains one or more full paper on computer programming during their 1st/2nd semester. Usually, such a course has a massive number of registered students belonging to various disciplines. Manual assessment of the lab assignments for such a course is a colossal task which requires huge resources especially large number of instructors, substantial no. of human effort and time. It is also very much difficult for the instructor to detect plagiarism and avoid submission of unwanted/unsolicited program. The software reported in this paper carries out this task automatically on behalf of the lab instructor. Performance of the software is verified with the help of a large number of programming assignments individually and in integration with LMS. Experimental results shows 100% accurate performance by the software.**

***Index Terms*—Assignment category, computer assisted, integrated, plagiarism detection, security, web based.**

I. Introduction

The expansion of engineering education in the recent past has posed new challenges to the academic community. IT and ITES industry has grown in many fold in recent years in our country [1]. Technology driven education is gaining momentum and the teaching-learning system is fast adapting itself to innovative use of technology. The present paper deals with the problem of verification and evaluation of huge number of computer programming assignments in a standard engineering/ polytechnic colleges as well as vocational training Centre. The number of students associated with computer programming classes' has increased very rapidly [2], [3]. So the proper evaluation of these programming assignments becomes more and more difficult for the lab instructors in a specific time frame. Several systems for checking programming assignments have been proposed earlier to reduce the workload of the lab instructor. Automatic submission in an evolutionary approach to computer science teaching community has been proposed to resolve this issue [4]. Students in programming courses may learn from peer evaluation of their work. Moreover a student may learn by evaluating other students' works and may find many useful techniques in them to improve his/her programming skill through collaborative learning [5]. A system was introduced with much more extra features to render online submission of assignments in the networked environment securely [6]. M. Joy and M. Luck have introduced BOSS a system for online submission and automatic testing of programming assignments for any specific programming language [7]. However, this system was unable to show the results automatically and not a generic in nature. The instructors are required to see the result manually. Reference [8] describes an automatic assignment checking system, with capacity for online submission and immediate evaluation of programming language assignments for a specific language by M. Amelung, M. Piotrowski and D. Rosner**.** The boss online submission and assessment system [9], [10], supports coursework assessment through collection of submissions, performing automatic tests for correctness and quality, checking for plagiarism, and providing an interface for marking and delivering feedback. Reference [11] shows BOSS2 that improves the security features of previous BOSS system and also concentrate on the object oriented view. Plagiarism in programming assignments can also be detected by monitoring different instances [12]. Reference [13] shows another system that helps users to submit assignments in secured environment. Reference [14] compares several automated assessment and plagiarism detection system. Some different assignment checking system shows in [15]-[19]. But the previously introduced systems are used for checking one or a few specific programming language assignments. Reference [20] describes a system which can be used for submission and automatic immediate assessment of programming assignments by creating executable file.

In this paper, we propose an automated system for assessment of programming lab assignments. Here the learners are required to upload their executable as well as the source code. The files are checked for malware and viruses at the time of submission. Test cases and input sets are defined and fit into the system by the instructor. The system executes the program automatically or with the help of instructors (only for graphical assignments) and stores the result in the database. The system also has a provision to detect plagiarism based on the line number, and the variables used in the system. Furthermore, learner can upload their source

Manuscript received September 24, 2012; revised October 30, 2012.

J. Mahata is with the Dept. of Information Technology of MCKV Institute of Engineering, Kolkata.

A. Mahata is with the Dept. of Information Technology of The Orissa Minerals Development Co. Ltd (OMDC) (e-mail: ashutosh.mahata@gmail.com).

S. Roy is with the Dept. of Computer Science & Engineering of National Institute of Technical Teachers' Training and Research (NITTTR), Kolkata.

DOI: 10.7763/IJEEEE.2012.V2.162

code file which also in tern be used to detect plagiarism. Report card is generated based on the result of the correctness as well as outcome of the plagiarism detection.

The structure of the rest of the paper is as follows: Section II presents the Overview of the proposed system. Section III provides the design views. In Section IV, we discuss various user interfaces to the system as well as the technology used to develop the system. Security features of the system are described in section V. Experimental results are reported in Section VI. Finally, conclusions are drawn in Section VII.

II. System Overview

This section provides the overview of the system from a functional point of view. There are three kinds of users for the system.

1) The Learner
2) The Instructor
3) The Administrator

Each of these individuals has a different roles and functionalities as far as the use of the system is concerned. A static overview of the system regarding these three categories of people as the actors and the related use cases is shown in the Fig.1 using UML notations [21].

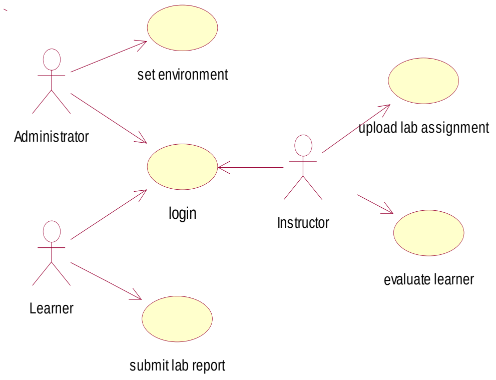


Fig. 1. Static overview of the system as a Use Case Diagram

These three actors have been assigned different roles in the system. Depending upon the role different menu is opened for different user. After login into the system the Administrator can add, delete and update any user information if necessary. The Instructors log into the system and post the assignment with specific assignment no. and will set the input and respective output. The learners will log into the system and upload the program i.e. the source code and executable file developed by them. The assignments will automatically executing for non-graphics assignments and the instructor will choose the graphics menu and run the executable files submitted by learners'. He checks the output whether it is correct or erroneous and accordingly marked the result. The Instructor has the option to check the source code to detect plagiarism.

The roles played by the actors mentioned above are summarized in Table I.

Based on the activities of the actors shown in Table I accordingly the following use cases are identified,

1) Login
2) Set Environment
3) Upload Lab Assignment
4) Submit Lab Report
5) Evaluate Learner

TABLE I: Summary of Activities Associated with Actors

#	Actor	Activities
1.	Learner	1. Login to the system 2. Submit the program developed by him/her by uploading into the system
2.	Instructor	1. Login to the system 2. Post the assignment 3. Add test cases and the respective output 4. Set submission deadline 5. Run graphics programs submitted by learners and check the output. 6. Check the source code if necessity occurs.
3.	Administrator	1. Login to the system 2. Set the subject code 3. Add new user 4. Delete existing user 5. Update user information

The use cases associated with the respective actors is shown in Table II.

This is shown in the Fig. 1.

TABLE II: Actors and the Associated Use Cases

#	Actor	Use Case
1	Learner	1. Login 2. Submit Lab Report
2	Instructor	1. Login 2. Upload Lab Assignment 3. Evaluate Learner
3	Administrator	1. Login 2. Set Environment

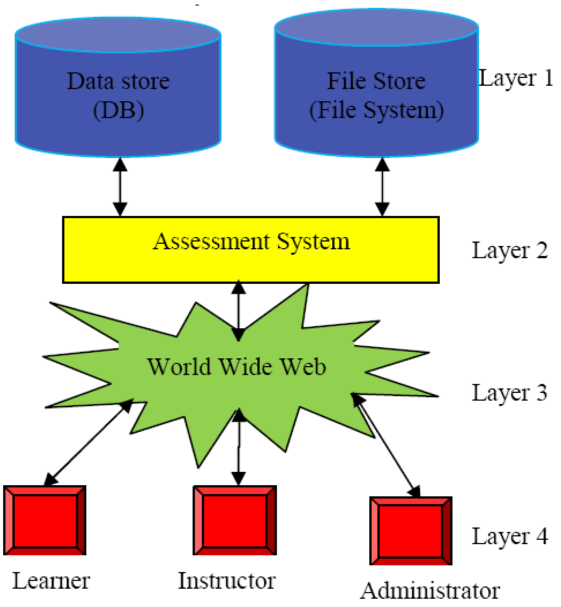


Fig. 2. High level architecture of the system.

Fig. 2 shows the high level architecture of the system. There are four layers in the system. Among them Layer1 is the combination of database as well as file system layer. The details of the users, the assignment details and the results are stored in the data store whereas the source code file and the executable file are stored in the file store. Layer2 is the Assessment system itself which calculate the result of non-graphical program helps the instructors to run the graphics program and then integrate the graphics programming output with the system generated output and generate report card. Layer3 is the web environment where the system will execute and Layer4 is the User Interface layer where the users can communicate with the system.

III. System Design

This section describes the design view of the system which includes the categorization of assignments, the architecture of the proposed system, and the deployment diagram of our proposed system.

A. Program Categorization

Categorizations of all assignment into different assignment category are the building block of this entire software system. All programming assignments are divided into two board categories graphical and non-graphical. The programming assignments which belong to non-graphics are divided into 5 categories. Irrespective of the programming language, we tried to fit each and every programming assignment into one of these five categories. At the time of assignment creation instructor/administrator assigned the proper category of any particular assignment and their submission deadlines. Assignment submitted within the prescribed date is defined as normal submission whereas submission beyond some permission date is defined as soft deadline. Beyond soft deadline hard deadline is defined, beyond which submission is not possible. Whenever assignments are executed particular block of code getting executed based upon the category of the assignment. Execution to report card generation system will automatically take care. Program categorizations are summarized in Table III.

TABLE III: Assignment Categories

#	Assignment Category	Description	Example Program
1	One_to_One	One Input, One Output	Program to determine wheather a number is even or odd
2	One_to_Many	One Input, Many Output	Program to find out the Factors of a number
3	Two_to_Many	Two Input, Many Output	Program to find out all the Prime numbers in between a given range
4	Many_to_One	Many Input, One Output	Program for Searching
5	Many_to_Many	Many Input, Many Output	Program for sorting
6	Graphics		Program to draw a circle

B. System Architecture

The inputs to the system are the solution files (executable file and source code) uploaded by the learner and the predefined set of inputs and outputs are created by the instructors/administrator at the time of creation of assignment. All the files are verified by the system for malicious infection and then rejected/stored in the database. The result of the graphical assignments is needed to mark to the system by the instructor. After execution report card is displayed individually to the instructors/administrator and to the students.

The architecture of the proposed system is illustrated in Fig. 3. For graphical programming assignments the instructors will run the executable files one by one and marked the result to the system which is stored in the data store. For other category assignments the system generates the output by running the executable file with predefined input sets and comparing system generated the output sets with the predefined output sets. Source code is tested by the system to detect plagiarism. Firstly, the system integrates these output sets of graphical, non-graphical assignments and plagiarism detection and generates report cards to the instructors and the students.

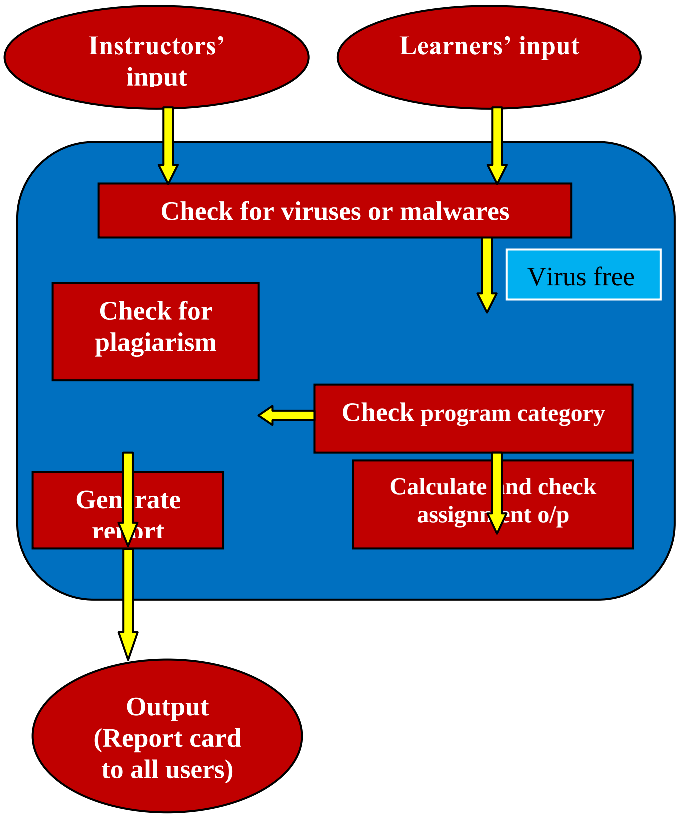


Fig. 3. Architecture of the system.

IV. System Developed

As far as the technology is concerned, cutting edge technology is used which is affordable and easy to learn and develop the software. Keeping in view of the pricing and licensing charges and all, it is developed free by using free version of the programming language and database of reputed vendors. Some limitations are there in developing any software in a free version of technology but as long as no of participant is not very huge it will never felt. The system is implemented with the ASP.NET using VB.NET Express Edition as the front end and SQL Server 2005 Express Edition as the back end database. There are three types of user of the system i.e. administrator, instructors and learners. After login into the system separate menu option is opened for different users depending upon their roles and responsibilities. Administrator has absolute control to the overall system. She/he can update the user list by adding and deleting user from the system and by updating the user information. She/he will assign permission to the instructors and learners to enter into the system by providing login id and password. In the assignment entry form, the instructor can update an assignment and can select the assignment category and set the submission deadlines. In this form, the instructors can set the test cases i.e. the input and output for that particular assignment. The learner can view the assignment list with details information of the assignments. They will solve the assignments independently and upload the executable file as well as the source code. The system will check the files for malware during upload. The learner can submit the assignment one by one by inserting the verification code at the time of each submission. The users' information, assignments details and the resultant outputs are stored in the Data store whereas the source code and the executable files are stored in the File store. The instructors have to click the graphics hyperlink to see the output of the graphics program. Instructors will run the graphics program by selecting assignment no and clicking on the submit button. The executable file will run and output is displayed to the instructors. After that instructors will enter their comments whether the output is ok or not. The result of the graphics programming assignments is then integrated with the system generated result of other category assignments. The system will check the source code for all the assignment for all category assignments. The result of graphical and non-graphical assignments as well as the result of plagiarism is reflected in the report card. The Instructors and the administrator can generate and assess the report for each assignment and also can view the source code. They also have the capability to generate report card for each individual learner. The learner can view the report card with the system generated feedback. The administrator, instructors and learners will able to know the assignment submission status on regular basis.

V. Security and Plagiarism Detection

Web security is a major concern in today's internet based technology driven community. The proposed system is protected by two labels of securities which guard the system from unauthorized user and malicious file. The administrators have the overall control to the system that provides the login id and password to the registered instructors and learners. After successful registration, the users i.e. instructors and learners have to wait for the login id and password. Only by using valid login id and password users can log in to the system. The system also protected with a strong antivirus installed in the server side. Which scan each and every program at the time of submission. Only malware/virus free files can be stored in the file store and malicious files are deleted by the system and error message will be displayed to the user.

```
// check for viruses
If (virus free) then
Files are uploaded and saved into the file store
Else
Files are deleted && send error message.
End
```

For each submission, the learners have to input the different verification code generated by the system. Here the verification code is the result of a mathematical function which is used to present auto submission. The system will generate the function and the learners have to solve it and the put the resultant value in the respective field.

The system can generate wrong report card due to plagiarism. Plagiarism for computer programming assignments can be divided into two categories.

1) Plagiarism by copying program from others
2) Plagiarism by using wrong logic

This assessment system has the capability of detecting both categories of plagiarisms.

Algorithm for detecting first category of plagiarism: This can be done by checking the no. of lines and variables used in the program. If the line numbers of two programs submitted for the same assignment is same as well as more than 80% of variables have the same name then the system will decide that the program is copied from others and the files are not evaluated and not saved in the file store.

```
// Check for plagiarism
If (same assignment submitted previously by other users)
{
If (no. of lines in submitted prog. =no. of lines in the previous program) then
{
        If (no. of variables in submitted prog. >=80/100*no. of variables in the previous prog.) then
        {
          If (variable names are same) then {
                Plagiarism detected
                }
        Else{
        Not detected
        File saved and executed (for executable file only)
        }}}
Else{
        File saved and executed (for executable file only)
}}
```

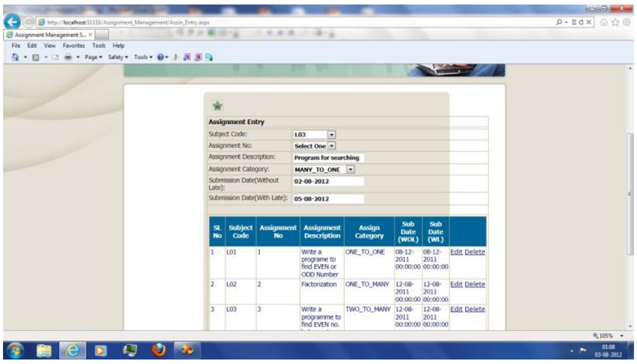
In 2nd category plagiarism the learners use the wrong logic in the program so that the program will produce the correct output as the original output but the logic is not written as per the assignment. For example, there is an assignment for sorting by using quick sort. If someone solves it by using bubble sort algorithm and then uploads the executable then the system will upload the same and marked it as correct one. To detect this type of plagiarism the administrator and respective instructors have the provision to open the source code file from the file store to detect plagiarism.

VI. Experimental Results

This section gives the experimental result and sample screenshot of the system. The system will marked an assignment as a correct one if its output is same as the instructors predefined output and plagiarism is not detected for the program. The system's decision is summarized in Table IV.

TABLE IV: RESULT SUMMARY

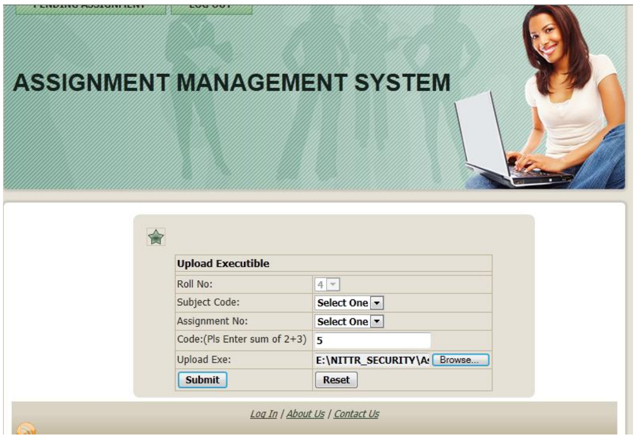
#	Program type (correct/incorrect)	System's decision	Result of plagiarism	Final Result
1	Correct	Correct	Yes	Error
2	Correct	Correct	No	Ok
3	Incorrect	Incorrect	Yes	Error
4	Incorrect	Incorrect	No	Error


Assignment Entry

Subject Code:	L03
Assignment No:	Select One
Assignment Description:	Program for searching
Assignment Category:	MANY_TO_ONE
Submission Date(Without Late):	02-08-2012
Submission Date(With Late):	05-08-2012

SL No	Subject Code	Assignment No	Assignment Description	Assign Category	Sub Date (WOL)	Sub Date (WL)	
1	L01	1	Write a programe to find EVEN or ODD Number	ONE_TO_ONE	08-12-2011 00:00:00	08-12-2011 00:00:00	Edit Delete
2	L02	2	Factorization	ONE_TO_MANY	12-08-2011 00:00:00	12-08-2011 00:00:00	Edit Delete
3	L03	3	Write a programme to find EVEN no.	TWO_TO_MANY	12-08-2011 00:00:00	12-08-2011 00:00:00	Edit Delete

Fig. 4. Screenshot to show how to fix assignment category

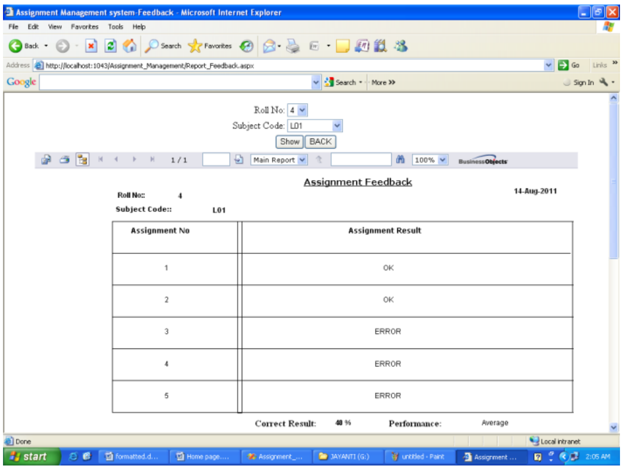


Upload Executible

Roll No:	4
Subject Code:	Select One
Assignment No:	Select One
Code:(Pls Enter sum of 2+3)	5
Upload Exe:	E:\NITTR_SECURITY\A: Browse...
Submit	Reset

Fig. 5. Screen shot for upload solution files by learners.

Fig. 5 shows how the learners are submit their solutions. Sample report card with feedback which is generated to the learner is shown in Fig. 6.


Assignment Management system-Feedback - Microsoft Internet Explorer

Address http://localhost:1043/Assignment_Management/Report_Feedback.aspx

Roll No: 4

Subject Code: L01

Assignment Feedback

14-Aug-2011

Roll No:: 4

Subject Code:: L01

Assignment No	Assignment Result
1	OK
2	OK
3	ERROR
4	ERROR
5	ERROR

Correct Result: 40 % Performance: Average

Fig. 6. System generated report to the learner.

The software is tested for a large no of correct and incorrect both categories of programs where some cases plagiarism are there. But for all the cases the system will produce correct result.

Fig.4, Fig. 5 and Fig. 6 represent the sample screenshots of the system. Fig.4 shows the screenshot of upload assignment problems where the instructors (or administrator) are fix the assignment category and upload assignment statement.

VII. Conclusion

A software system that will automatically execute lab assignments for graphical & non-graphical and send feedback to the concerned individuals has been proposed and presented in this paper. It is basically a web based system which will expedite the process of submission of lab assignment online in a safe, secure and confidential manner. The system is secured by the two levels of security systems which reduces the probability of system break down. The system accepts executable files so that the students are free to implement their programs in any programming languages. The proposed system has the potential for becoming comprehensive student management system. This software is tested for a large number of programming assignments of all categories with hundred percent accurate results. This has the capability for seamless integration with any Learning Management System (LMS).The system efficiently handles large no of programming assignments, simultaneously. It reduces the work pressure of the instructors and save valuable times of all stakeholders. Students need not have to take printout for submitting the assignment which effectively saves their money, time and workloads. Millions of pages are used to take printout of the programming assignment. Submission through this system reduces it to zero. This way it may help to achieve green computing also. Experimental results show 100% effectiveness of the proposed system.

References

[1] H. S. Lohit, "Indian IT Industry: Current Status and Challenges," presented in Industry – Academia Workshop, Lucknow, April 2005.

[2] A. Blom and J. Cheong., "Governance of Technical Education in India: Key Issues, Principles, and Case Studies," World Bank Working Paper, no. 190, March 2010.

[3] S. K. Singh, "Information Technology in India: Present Status and Future
Prospects for Economic Development," Directions, vol. 7, no. 4, pp. 5-14, 2006.

[4] M. Luck and M. Joy, "Automatic Submission in an Evolutionary Approach to Computer Science Teaching," Computers and Education, vol. 25, no. 3, pp. 105-111, 1995.

[5] A. Trivedi, D. C. Kar, and H. Patterson-McNeill. "Automatic assignment management and peer evaluation," Journal of Computing Sciences in Colleges, vol. 18, no. 6, pp. 151-164, June 2003.

[6] M. Luck and M. S. Joy, "A Secure On-Line Submission System," Software - Practice and Experience, vol. 29, no. 8, pp. 721-740, 1999.

[7] M. Joy and M. Luck, "On-line Submission and Testing of Programming Assignments," Innovations in Computing Teaching, ed. J. Hart, pp. 97-103, SEDA,1995.

[8] M. Amelung, M. Piotrowski, and D. Rösner, "Large-scale Computer-Assisted Assessment in Computer Science Education New Possibilities, New Questions," in Proc. International Workshop on Blended Learning, Edinburgh, United Kingdom, 2007.

[9] M. Joy, N. Griffiths, and R. Boyatt, "The boss online submission and assessment system," ACM Journal on Educational Resources in Computing (JERIC), vol. 5, no. 3, September 2005.

[10] M. Joy, P. S. Chan, and M. Luck, “Networked Submission and Assessment,” in Proc. the 1st Annual Conference of the LTSN Centre for Information and Computer Sciences, LTSN-ICS 2000.

[11] M. S. Joy and M. Luck, “Computer-Assisted Learning using the Web,” in Proc. the 5th Annual Conference on the Teaching of Computing, Dublin, pp. 105-108, 1997.

[12] M. S. Joy and M. Luck, “Plagiarism in Programming Assignments,” IEEE Transactions on Education, vol. 42, no. 2, pp. 129-133, 1999.

[13] D. C. Kar, “Detection of plagiarism in computer programming assignments,” Journal of Computing Sciences in Colleges, vol. 15, no. 3, pp. 266-276, 2000.

[14] P. Ihontola, T. Ahoniemi, V. Karavirta, and O. Seppala, “Review of recent systems for automatic assessment of programming assignments,” in Proc. the 10th Koli Calling International Conference on Computing Education Research, pp, 86-93, 2010.

[15] N. Zamin et al., “WAGS: A Web-Based Automated Grading System For Programming Assignments From Users’ Perspectives,” in Proc. International Conference on Science & Technology: Application in Industry & Education, Malaysia, 2006.

[16] A. Kurnia, A. Lim, and B. Cheang, “Online Judge,” Computers & Education, vol. 36, no. 4, pp. 299-315, 2001.

[17] L. R. Chien, D. J. Buehrer, and C. Y. Yang, “Using Test-Driven Development in a Parse Tree Based Online Assessment System,” in Proc. IADIS International Conference e-Learning, 2007.

[18] M. Choy, U. Nazir, C. K. Poon, and Y. Yu, "Experiences in Using an Automated System for Improving Students' Learning of Computer Programming," in Proc. ICWL, pp.267-272, 2005.

[19] B. Cheang and A. Kurnia, “A. of programming assignments in an academic institution,” Computers & Lim and W.C. Oon,” Automated Grading Education, vol. 41, no. 2, pp. 121-131, September 2003.

[20] J. Mahata, A. Mahata, and S. Roy,” Web Based Computer Assisted assessment of Programming Lab Assignments (graphical and non-graphical)”, in Proc. National conference on Emerging Trends in Educational Informatics, February 16-17, 2012.

[21] G. Booch, J. Rumbaugh, and I. Jacobson, The Unified modeling language User Guide, Pearson Education, Asia, 2002.

Jayanti Mahata was born in 1983 in a native village of West Bengal, India. After graduating with Bachelor of Technology in Computer Science and Engineering from the Vidyasagar University in 2004, she starts her career as a lecturer in Computer Science and Technology of Dumkal Polytechnic, West Bengal, India. She leaves the job and finished her M.Tech with specialization Multimedia and Software System from National Institute of Technical Teachers’ Training and Research, Kolkata (Under West Bengal University of Technology) in 2007. Mrs. Mahata is presently attached to the Dept. of Information Technology of MCKV Institute of Engineering, Kolkata, as Assistant Professor .She has taught various topics of Computer Science at diploma and undergraduate level at various institutes for the last 6 years. She has published many papers in various international and national conference proceedings. Her areas of interest include Educational Informatics, Multimedia Technology and Distributed Computing. She is life member of Computer Society of India and Internet Society.

Ashutosh Mahata is presently associated with Dept. of Information Technology of The Orissa Minerals Development Co. Ltd (OMDC), A Govt. Of India Enterprise, Kolkata, as a capacity of Manager (IT). After graduating with Bachelor of Technology with specialization in Information Technology from the University of Kalyani in 2003, he joined CMC Ltd. (A TATA Enterprise) as a Graduate Apprentice. Gradually, he was promoted to IT Engineer with a role of the Software Project Manager. He has completed various projects as a Software Developer, Database Administrator (DBA) and Project Manager for the last eight years associated with CMC Ltd. He is the Oracle Certified Professional (OCP) in Oracle 9i Developer Track. He has completed his Master of Business Administration with specialization in Information Technology from Sam Higginbotom Institute of Agriculture, Technology & Sciences (Formerly Allahabad Agricultural Institute-Deemed University) in 2012. He has published many papers in various international and national conference proceedings. His areas of interest include Educational Informatics, Database Management, Project Management, and Enterprise Resource Planning.

Samir Roy is presently attached to the Dept. of Computer Science & Engineering of National Institute of Technical Teachers’ Training and Research (NITTTR), Kolkata, as Associate Professor. After graduating with honours in Physics from the Presidency College under University of Calcutta, he obtained his B.Tech (1989, University of Calcutta), M.E (1991, Jadavpur University) and Ph.D (2004, University of Kalyani) in the field of Computer Science & Engineering. He has taught various topics of Computer Science at undergraduate and post-graduate level at various institutes for the past 19 years. He has around fifty articles in various international and national journals and conference proceedings. His areas of interest include Educational Informatics, Artificial Intelligence, Soft Computing, Theory of Computation.