

TOWARDS PERSONALIZED PEER-MATCHING FOR LEARNING PROGRAMMING

By:

Joseph Sunday

Muhammad Usman Ali

Supervisors

Dr. Ghislain Maurice Norbert Isabwe

Ms. Renée Patrizia Schulz

This master's thesis is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education. However, this does not imply that the University answers for the methods that are used or the conclusions that are drawn.

University of Agder, 2016

Faculty of Engineering and Science

Department of Information and Communication Technology

Abstract

Peer review has been used as a proven process for sharing knowledge and improving skills among the learners. This became more popular by the use of ICT. Peer review provided effective, qualitative and summative feedback. It is challenging when it comes to the peer review of programming languages where there is no specific single solution to the problem. Programming languages students need peer review to learn new techniques while sharing their ideas to solve the problems they face. This helps the programming language teachers in the marking process of the assignments and hence minimizes their workload.

Students receive feedbacks from reviewers which helps them to learn the concepts from different perspectives by looking at others' work and techniques. The main challenge becomes the students' needs in different sub-sections of the problem solving. In this thesis, we used the personalized need for review by introducing a personalized peer matching algorithm. It is used to assign reviewers to students based on their personal needs and performances.

Peer matching is used in combination with the assessment criteria that are used in the assessment of the submitted solutions. This helps the student get the most needed help on the assessment criteria where s/he is weaker compared to others. This is done by assigning reviewers that are not on the same performance level on those criteria. The algorithm groups the students according to their performances in three levels (low, medium and high). The algorithm assigns each student with three reviewers one from each level. These students must have scored different performances on the same criterion.

In order to use the algorithm and show its benefits to students, we developed a peer review system for programming language based on teachers' and students' perspectives. This system uses peer matching algorithm on an introductory programming course being offered at the University of Agder with the first year students of computer science. Students were given a programming problem to solve and submit the solution for review. Students were assigned reviewers using the personalized peer matching algorithm and they reviewed others' solutions. The algorithm showed significant benefits to the students as they received improving feedback on criteria they had not performed well in their previous assignment.

Keywords: Peer review, collaborative learning, personalized learning, peer matching, programming assessment criteria

Acknowledgement

Firstly, we would like to express our deepest appreciation to our supervisors Dr. Ghislain Maurice Norbert Isabwe and Ms. Renée Patrizia Schulz for encouraging us to complete this thesis. Our efforts for this thesis would have not been possible without their motivation, guidance as well as continuous support. We could not have imagined having better supervisors and mentors for our masters' thesis.

We would like to extend our gratitude towards our families for their constant encouragement which helped us in completion of this work. Last but not the least, our special thanks also go to our fellows in developing the project and all the people of UiA who have willingly helped us out with their abilities.

Joseph Sunday and Muhammad Usman Ali

UiA Grimstad

16 May 2016

List of figures

| | |
|---|----|
| Figure 1: T-FLIP architecture source [11] | 3 |
| Figure 2: Problem solving steps [37] | 15 |
| Figure 3: Human-centred design process [46] | 24 |
| Figure 4: Previous experience of peer review | 28 |
| Figure 5: Benefits for students from peer review system | 28 |
| Figure 6: Peer review by anonymous or known identity | 29 |
| Figure 7: Students' grouping | 29 |
| Figure 8: Feedback consideration | 30 |
| Figure 9: Reviews per assignment | 30 |
| Figure 10: Using meaningful data types and variable naming | 30 |
| Figure 11: Correct use of syntax and usage of If statements | 31 |
| Figure 12: Usage of loops and classes | 31 |
| Figure 13: Using methods and applying comments | 31 |
| Figure 14: Check correctness and relevant lines of code | 32 |
| Figure 15: Coding style and code readability | 32 |
| Figure 16: Use case diagram according to Volere template [47] | 38 |
| Figure 17: Peer matching algorithm | 42 |
| Figure 18: Students performance levels and reviewers distribution | 43 |
| Figure 19: Peers matching process | 46 |
| Figure 20: First iteration of interface design | 49 |
| Figure 21: Second iteration of interface design | 50 |
| Figure 22: Third Iteration-information page | 51 |
| Figure 23: Third Iteration-review process page | 51 |
| Figure 24: Conceptual model for peer review | 56 |
| Figure 25: Peer review process architecture | 57 |
| Figure 26: Add new course to the system | 59 |
| Figure 27: Add an assignment to DAT001 course | 60 |
| Figure 28: Teacher view on students review and feedback | 61 |
| Figure 29: Feedback given to a student by three reviewers | 62 |
| Figure 30: Review process | 65 |
| Figure 31: Feedback from reviewer 1 | 66 |
| Figure 32: Post-survey on the difficulty of peer review process | 69 |
| Figure 33: Post-survey on learning from reviewing process | 69 |
| Figure 34: Post-survey on reviewer distribution | 70 |
| Figure 35: Post-survey on learning from feedback | 70 |
| Figure 36: Post-survey on assessment criteria usefulness | 71 |
| Figure 37: Post-survey responses for interface navigation | 71 |
| Figure 38: Assignment task | 84 |
| Figure 39: Assignment solution | 85 |
| Figure 40: An added course | 86 |
| Figure 41: Course related assignments | 86 |
| Figure 42: Adding a criterion to assignment | 86 |
| Figure 43: Submitting assignment solution | 86 |
| Figure 44: Assignment sample to peer review system | 87 |
| Figure 45: Submitted reference solution | 87 |

List of tables

| | |
|--|----|
| Table 1: Five ways of experiencing learning programming [38] | 15 |
| Table 2: Measures to assess students [41] | 16 |
| Table 3: Roles of system users..... | 41 |
| Table 4: Assessment criteria [44] | 47 |
| Table 5: Students in levels with their corresponding scores | 53 |
| Table 6: Assessment criteria with corresponding weights..... | 54 |
| Table 7: Students in their respective performance levels | 62 |
| Table 8: Students assigned with reviewers | 63 |
| Table 9: Students in levels with their corresponding scores..... | 64 |

List of abbreviations and acronyms

| | |
|---------|--|
| A-PASS: | Agder Peer Assessment System |
| CSCL: | Computer Supported Collaborative Learning |
| HCI: | Human Computer Interaction |
| IBM: | International Business Machines Corporation |
| ICT: | Information and Communication Technology |
| ISO: | International Standard Organization |
| IT: | Information Technology |
| LSA: | Latent Semantic Analysis |
| Mac: | Macintosh |
| MGR: | Maps and Geospatial Revolution |
| MOOC: | Massive Open Online Course |
| PC: | Personal Computer |
| PCR: | Peer Code Review |
| PL: | Personalized Learning |
| PRISM: | Peer Review Managements System for Programming Assignments |
| RBAC: | Role Based Access Control |
| T-FLIP: | Test Driven Skills Learning Programming |
| UiA: | University of Agder |

Contents

| | |
|--|-----|
| Abstract | i |
| Acknowledgement | ii |
| List of figures | iii |
| List of tables | iv |
| List of abbreviations and acronyms | v |
| 1. Introduction | 1 |
| 1.1 Background | 1 |
| 1.2 Collaborative learning | 4 |
| 1.2.1 Social constructivist learning theory | 5 |
| 1.2.2 Computer supported collaborative learning | 5 |
| 1.3 Thesis outline | 6 |
| 2 State – of – the – art | 7 |
| 2.1 Computer supported assessment systems | 7 |
| 2.2 Peer assessment | 9 |
| 2.3 Peer assessment of programming course | 11 |
| 3 Problem statement and research questions | 19 |
| 4. Methodology | 22 |
| 4.1 Data collection on assessment of programming courses | 22 |
| 4.1.1 Interviews with teachers | 22 |
| 4.1.2 Survey on students' perspectives | 23 |
| 4.2 Data analysis | 23 |
| 4.3 Human-centred design | 23 |
| 4.4 User testing and evaluation | 25 |
| 5 Requirements engineering | 26 |
| 5.1 Summary for teachers' interview | 26 |
| 5.2 Students' survey data analysis | 27 |
| 5.3 Requirements | 33 |
| 5.4 Peer matching algorithm | 41 |
| 5.5 Random subproducts algorithm | 44 |
| 5.6 Assessment criteria | 47 |
| 5.7 Interface design | 48 |
| 5.8 Personalised learning | 51 |
| 6. Implementation and validation of peer review system | 55 |
| 6.1 Design phase | 55 |

| | | |
|-----|---|----|
| 6.2 | Implementation..... | 59 |
| 6.3 | Validation | 66 |
| 7. | Discussion and limitations | 73 |
| 7.1 | Discussion | 73 |
| 7.2 | Limitations | 75 |
| 8. | Conclusion and future work..... | 76 |
| 8.1 | Conclusion..... | 76 |
| 8.2 | Future work | 77 |
| | References..... | 78 |
| | Appendices..... | 84 |
| | Appendix A: Assignment task and solution submitted to T-FLIP..... | 84 |
| | Appendix B: System interfaces..... | 86 |
| | Appendix C: Post-user testing survey for peer review system | 87 |
| | Appendix D: Peer matching code | 88 |

1. Introduction

1.1 Background

Higher education is based on the skills acquired from school where teachers with high academic skills teach students and make them familiar with the real world tasks [1]. Acquiring these skills by students has been difficult in some cases, where the courses they are taking may need further thinking beyond students' capability levels. These courses include programming courses where everyone must be using his/her own understanding and reasoning hence it needs everyone's logic to solve a given programming task. This makes it difficult for students to learn programming [2] from teachers' reviews only and they need other assistances to learn the course content. Recruiting many teachers for the same course can be expensive and students really need help in the programming courses, although there are students that are more skilled with programming knowledge than others. Students can help their fellows even though it is time consuming and unplanned sometimes.

Students with more programming skills are helpful in the learning process of programming language courses when they are well organized. These students can be brought together through peer review process and give feedbacks to their fellow students. Students need reviews for their works in order to get more knowledge of how they could have solved the given tasks. Also by reviewing others' work, students get more ideas and techniques for solving the task. There is a need for the feedback from which they acquire more knowledge and skills for programming and these skills can help them for their future tasks [3].

Apart from the feedback needs, there is a need for a learning and teaching method that is reliable and affordable for the educational institutions to provide quality education to the students. This method can be obtained by engaging students who are directed to make their fellows learn while getting more skilled through peer review. This method can reduce teacher's time spent with students even if they just have to guide the students about review process and make a follow-up to the method used in the process.

Another focus for this project will be the approach used in the review process where reviewers will be able to review with minimum degree of friendliness feelings or the radicalness [3]. Reviewers will be directed by the marking scheme to improve reliability of the reviews. The marking scheme or reference document will be prepared by the teacher responsible for the course assignment. This marking scheme (reference document) is the solution for the task given

for the corresponding assignment, containing one possible method that can be used to solve that assignment. It will be used only in that particular review process and its feedback.

There has been peer review from early ages [4] but the way reviewers are assigned is not motivating as most of the review assignments were done randomly from the students or picking the most skilled [5] and making them the reviewers for others. Students need to receive reviews from all possible performance levels; here we can say, the low performant, medium and the high performant students. These techniques are helping the strong students to evaluate and learn from reviewing while the authors will only learn from feedbacks instead of learning from the strong students' solutions too. There should be a technique that help students to get reviewers from all performance levels and also where all the students are reviewers. This will help them view what others did for the same task and reflect to their own task solutions so that they can make corrections. Peer review is highly recommended because it reduces teachers' workload [6] and increases learning outcome [7].

Peer review as a way for students to judge their fellows' work and provide them with feedbacks [8], students need to be guided on the assessment process [9]. This can be done by providing the assessment criteria to the reviewers before they are conducting the review process. These criteria are set by the teacher and must be relevant to the requirements of the deliverable task. But the problem raised is how to choose the assessment criteria and the amount of criteria that can be used in the process. Based on [10], the review process is free and open to students to provide feedback based on their knowledge and the format they want to use while giving the feedback to the assigned tasks. Free feedbacks are helpful but needs improvement and boundaries. Considering the limitations for the feedback, students (reviewers) should provide comments in the scope of the assignment.

Motivation

The idea of working on peer review for programming languages came into our minds while we were studying and needed to receive and to give support from/to our fellows. It was needed to improve the quality of our deliverables but because all of us were working on separate projects works, it could not be possible. This made us think of developing the idea of peer review in programming languages as a solution for computer science [3] students. Our core motivation was accelerated by the development of the test driven skills learning programming (T-FLIP) being developed by the University of Agder (UiA) in collaboration with other Norwegian universities. It is a learning management system that is used in the evaluation of programming

language courses' assignments. T-FLIP allows teachers to provide assignment tasks with reference solutions. Students submit their solutions and get compared to the teacher's solution. The system provides automated feedback about pass or fail (of a specific subject) to the students.

The system lacked the peer review part which is among the crucial parts that can help students to learn from their fellow students. We worked on this part as our contribution to the system, not only the development of the peer review part but the algorithm to be used in the peer matching as well as the assessment criteria for the peer review process. After completion of this feature, it can be integrated or adapted to the existing system in order to make it helpful for the students using T-FLIP. Figure 1 shows the T-FLIP architecture as below:

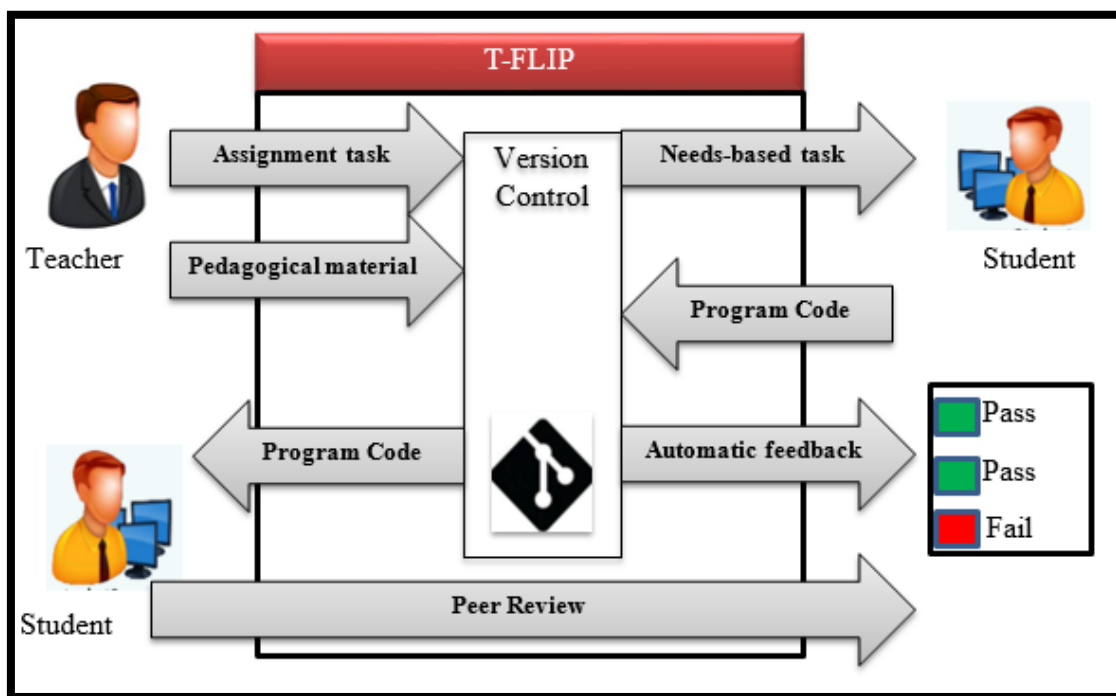


Figure 1: T-FLIP architecture source [11]

Figure 1 shows the functioning of the system architecture of the whole T-FLIP system. Teachers create tasks and save them. These tasks can be seen by students who can engage with the tasks and solve them. The solution is submitted to the system as program code. The student receives automatic feedback from the system which shows whether the student passed or failed the task. The student can redo the task and re-submit it to the system. As this report is being written, the system does all the mentioned tasks as discussed above. The remaining part is the peer review part which is the focus for our project where students will be able to review others' work by giving quantitative, qualitative or both types of feedback.

After having understood the basic mechanism of the reviewing process, we investigated the needed peer matching algorithm and the assessment criteria that will be used to evaluate programming language courses. These criteria are developed and used by teachers during the evaluation, so students can use them as orientation during the review process. Kari [12] mentioned general and common programming language topics which can be used as a reliable source to prepare an assessment criteria. Specifically, students need to know what the most important chapters for programming language courses are and this knowledge can be obtained from introductory programming courses.

To be able to do review of code guided by teachers' criteria, it is necessary that students obtain the most basic knowledge about these topics. They have to know what it meant by the criteria list and how to apply them. These topics will be used to generate assessment criteria for assessing introduction to programming course at Bachelor level at University of Agder in the development and validation of the peer review system. During this project work, we integrate both ideas from teachers teaching programming courses and students learning programming to get to know the most used topics in the course. The combination of the chapters introduced by Kari and the criteria used by teachers (that we found by conducting interviews) makes a basis for the assessment criteria for programming language courses, specifically, the introduction to programming course.

1.2 Collaborative learning

According to Dillenbourg in [13], collaborative learning is defined as the working together of more than one person to learn or to solve a problem. This is the main focus of this project where we are purposely interested to use the collaboration techniques to help students learn programming through peer review. Through collaborative learning, students are able to help their peers while learning from each other. Students are learners, and learners have tasks to solve. Sometimes, these tasks can be too difficult to solve, based on many factors like lack of specific knowledge, lack of guidance or lack of time, just to mention a few.

Peer review provides help which can make the students understand the tasks and solve them. Collaborative learning is more crucial while achieving critical thinking. According to Anuradha [14], individual learners are able to succeed with higher level of learning and acquire more information when they collaborate and work in groups than working individually. When students use collaborative learning, they are able to solve problems, share knowledge, build collaborative knowledge skills, which can lead to deeper understanding and learning.

1.2.1 Social constructivist learning theory

Constructivism inherits learning theories from cognitive and humanistic or social perspectives [15]. According to Atherton's constructivist theory research, opposing to behaviourism where the organism is treated as a black box, cognitive theory recognises the importance of the mind in making sense of the device presented. In behaviourism, the role of the student is primarily to assimilate whatever the teacher gives. Constructivism, specifically in its social form, suggests that students should be involved in solving the tasks. Students get highly involved in the learning process by engaging themselves in peer review (where all the students need to give, receive and share the knowledge they have among the peers). Teachers will be there to review and to cater any complaint that comes, but the most needed are the students. This helps students share their views on others' work and give advice about better problem solving techniques.

1.2.2 Computer supported collaborative learning

Computer supported collaborative learning (CSCL) according to Lipponen, is how collaborative learning supported by technology (using computers) can improve peer interaction and work together in groups, and how collaboration and technology help distributing and sharing of knowledge, skills and expertise among the peers [16]. CSCL is more grounded on different concepts of learning, research methodology, pedagogy, and research questions than its antecedents, computer assisted instructions, intelligent tutoring systems and logo-as-Latin did [17].

As technology enhances, education is among the crucial fields that is more supported by technology. With the use of technology, collaboration of learners has become more widely used, because of perishing barriers of time and location. Peer review supports teachers in marking the assignments given to the students, hence the students get the opportunity to learn from their peers. This makes students more confident and improves their decision making culture with an additional skills augmentation and participation [16] in their own learning. Learning science as a whole has extended from a narrow focus on individual learning to an incorporation of both individual and group learning [18]. The enhancement of CSCL is simultaneously contributing and developing both methods of learning (individual and group).

This research project aims at using peer review as collaboration learning method to enhance students' learning through a detailed and personalised feedback to help students improve their programming understanding as well as their skills.

1.3 Thesis outline

This section describes the outline of the thesis as follows; the first section is the introduction about the peer review as a process and what it takes to be academically helpful to the students. The second section is about the State-of-the-art of peer review, which consists of the related works. We used these related works to describe the weaknesses in the current peer review systems. The chapter “Problem statement and research questions” explains the problem itself and the reason for the research being conducted. The Methodology chapter describes the qualitative and quantitative research methods including interviews and surveys which are used for data collection. It provides a detailed explanation of the peer review system, its purpose, functionality and the expected outcome. The next chapter highlights requirement engineering along with the design specifications. Based on personalized peer matching, Personalized learning describes the detailed peer review process. The Implementation and validation of peer review solution explains how the designed prototype is implemented and validated. It further discusses its implementation according to the TFLIP data structure. Finally, we conclude and propose future directions.

2 State – of – the – art

The state of the art for this project consists of three main parts: computer supported assessment systems, peer assessment and peer assessment of programming courses.

2.1 Computer supported assessment systems

The use of computers has been proven to be an important educational and instructional tool. In any computer assisted teaching method, the main assistance and learning tool is the use of computer. If the teacher uses the computer appropriately and consciously, it positively affects the students' success [19] as well as minimizes the teacher's workload. The number of institutes providing higher education is increasing and therefore, the number of students who study at these institutes is also increasing. Using computer supported assessment systems can be beneficial in numerous ways. The advantages can be enhancing feedback to students and staff, time saving, reducing administrative loads and improving the balance of assessment methods [20].

In this section, we highlighted some of the successful computer supported systems in existence, which are used for educational assessment tasks. There have been many systems of this kind and they helped in the process of assessment for both: students and teachers. The students receive quick and exact feedback and teachers are assisted to perform their duties in less time compared to traditional assessment.

Firstly, T-FLIP [11] is a learning management system that aims to establish new learning technology for students taking programming language courses at the University of Agder. This is done by developing a learning management system closely linked to the programming tasks solving. It includes the evaluation of the assignments done by the students during programming courses.

While using T-FLIP, students work in an industry near programming environment. This prepares the students for situations that are similar to what they will undergo deal with at work after they graduate from university. T-FLIP has different components which are developed to provide a successful learning management system. These include electronic assignment hand-in, automatic evaluation and grading and automatic feedback.

T-FLIP allows students to get their assignments evaluated automatically after submission. Students run the evaluation function of the tool after submitting their assignments. The tool in response, provides automatic feedback with a pass or fail message depending on the submitted

solution. The tool compares the submitted solution with the solution provided by the teacher. The tool allows the student to re-submit the assignment solution whenever the fail message is received. This allows the student to correct the solution even though s/he has no any other guidance to follow for correcting the solution. This may be frustrating for the student who is submitting the solution and gets a fail message without any further assistance to correct the assignment solution.

For the high performing student, assignment redo and re-submission will be helpful as s/he can try to see multiple options of solutions which are possible. With all ideas in mind, we analyzed possible solutions for assistance. The peer review process allows students who are low performing to be assisted by the high performing students with the further guidance of the teacher's assignment solution. Additionally, provided assessment criteria supports the reviewers. Combining all these facts will make the tool more helpful and the students will get assistance for their solutions through reviews and through reviewing, since this makes them analyze other students' solutions.

Secondly, there is a tool for assessment of assignments using computer support called FrontScraper [21]. This tool allows the teachers to mark assignments in sub-assignments. These sub-assignments are divided based on the knowledge level of the students submitting the solutions. Also sub-assignments are grouped based on similar answers which are comparable in the assignment content. This tool was used in a bachelor course (Operating systems) at the University of Agder. The results showed that using the tool gave the same results as if the teacher would have done it traditionally with some minor difference depending on the algorithm used. The algorithm used in this assessment tool is entropy metrics' algorithm which cannot assess the relevance of the text used in the answer by the student [21]. The system is not grading but it helps teachers to organize assignments in an easy way to mark. The tool showed improvements in the marking process but also can be better if combined with a peer review component so that students can view and give feedbacks to their colleagues which can help them learn from others [3].

Another study about automatic grading systems using Latent Semantic Analysis (LSA) techniques was conducted [22]. This system was used to reduce the teacher's burden for marking assignments traditionally and it can support them in the marking process. The system generates assignment grades from the cosine similarity that showed the relatedness of students' assignments and the model answers obtained from the latent semantic vector space. It was used to grade programming language assignments. Although LSA was able to grade the assignments

consistently and faster, it was not able to detect programming orders and symbols. While complementing the human necessities in learning, we are not developing a grading system but we use a pre-grading idea of automated grading to initialize the sorting process which leads to peer matching. We combine peer matching with assessment criteria for the benefits of students learning programming.

2.2 Peer assessment

Peer assessment is an educational arrangement where students judge each other's performance quantitatively and/or qualitatively [23]. Peer assessment stimulates students to reflect, discuss, and collaborate in their learning process [10]. In the current education, there is an improvement in teaching students to learn from all the resources available. The first resource being the student himself together with his peers, he should be able to learn on his own. The main advantages of peer assessment include the following [24]:

- Improvement in learning; a result of making the student part of the learning environment and being self-responsible for his/her learning.
- Increased chances of students to continue learning through their lives.
- Enhanced ability to practice what they learn in their real lives.
- Multiple feedback is given to students for improvement purpose.

In order for the students to learn, they must be able to assess their knowledge capability and these capabilities must be shown by evaluative sources including assignments and teacher completed evaluations [25]. This is not only for some courses but it is the same for all subjects being taught at all educational levels. The helpful method of learning is the peer review where students assess their fellow students and give them either feedbacks or grades. This method has been used by several academic institutions to improve the learning capabilities.

Topping [10] defined peer assessment as an arrangement where individuals consider the amount, level, value, worth, quality, or success of the products or outcomes of learning of peers of similar status. He further discussed the solid foundation and methods involved in peer assessment. After reviewing the ongoing literature including process and the study outcome, it indicates that peer assessment benefits reliability and validity in various ways. The whole process of peer assessment has shown positive effects on students' achievements and attitudes. It proved that these effects are at least the same or better than the effects of assessment by the teachers. Although the evidence for such effects is so far more limited. The study concluded that the peer assessment using computers is apparently growing fast [10] , but it misses the

comparison between peer assessment and teacher assessment even though this is not our main focus.

In 2014, a research paper was presented about using assessment for Learning Mathematics with Mobile Tablet Based Solutions [26] for the system known as Agder Peer Assessment System (A-PASS). This paper was presented after the development of a tool that was used to help students to assess their fellows in a mathematics course. The system contains a web based tool running on mobile devices, where students after submitting the assignment can get a review from either individuals or a group. This tool supports only mathematics assignments and their review by using a canvas component integrated in the tool. This helps the reviewer to write on top of the original assignment submission using different colors to differentiate the original and the review. The tool was regarded useful by students because they could acquire more knowledge from their fellow students when they got feedback on their works.

Even though this was a success for the reviewing process, the algorithm used for assigning the reviewers was not promising for improvement. Reviewers were assigned using Fisher-Yate shuffle and a student could be reviewed by one with a lower capability. This can be not improving to the student being reviewed. This could have been more successful if the research included an algorithm to match peers based on their needs. The review process in the tool was based on the reviewer's knowledge to review the work based on his knowledge not based on pre-set corrections, and this could cause more bias on the reviewed student. The tool could have been more helpful if reviewer were given reviewing or assessment guidelines. This could have been more efficient and the results could reduce the teacher's reviewing time and students' complaints.

Another study was conducted to explore the effects and validity of online peer assessment in high schools and to analyze effects of different types of peer feedback on students. A total of 184 students of grade 10, developed their individual course projects by using online peer review assessment activities in a computer course where the students acted both as an author and a reviewer. The research results were evaluated by experts and peers, which indicated that the students have improved their project work progress being involved in the peer assessment activities. The results determined by the peers and experts were very similar, which demonstrates that the process of peer assessment at high school level is valid. The study further categorizes the peer feedback into 4 different categories: 1. Reinforcing, 2. Didactic, 3. Corrective and 4. Suggestive. The reinforcing peer feedback was considered as most beneficial for students in development, whereas the didactic feedback and corrective feedback might not

be very effective and play an unfavorable role for the improvement of students' projects. Similarly, the Suggestive feedback can be helpful in the initial activities of peer assessment activities but not very useful and effective in the later parts of peer assessment. [27] Online peer assessment proved to be cost effective and efficient, but the security and transparency issues were not discussed. The reviewers used in this research were both teachers and students which make it more complicated. It was more time consuming for teachers which could have been better to use only students for their own benefits.

A peer review study was conducted on Coursera MOOC where students worked on Maps and the Geospatial Revolution (MGR) task. [28]. It was a 5-week assignment where every student was told to choose a mapping technology platform to use. They were given the option to select a mapping story to work on which included the travelling experience, good location or hometown. After completion of mapping, the peer grading accounted four assessment criteria: clarity, convincingness, quality and aesthetic of the design. Students' final results were calculated based on the medians of the students' grades. Coursera's peer grading was considered not consistent compared to self-grading. The research discussed about missing the validity of the peer grading while designing a MOOC after finding the courses' peer grading lacking validity. After finding this useful information about the peer grading in Coursera, we realized that using assessment criteria in the evaluation of programming language assignments can be useful for the learnability.

2.3 Peer assessment of programming course

Satu and Kari [29] presented a simple student self-assessment tool that could be used to motivate learning and track progression in programming courses. The tool had a survey questionnaire that used Bloom's revised taxonomy [30] as a base of its scale. In their results, they found that the students could place their knowledge to the taxonomy based scale and found out that it helped in the improvement of their knowledge.

This study showed an improvement for the students using it but there was a lack for the assessment about the submitted work instead it was only based on the students' understanding which is not fair as it had no proof for the questionnaires answers.

Peer Review Managements System for Programming Assignments (PRISM) [31], a web based peer review management system, presented a research on the peer review for programming assignments that was done with senior-level students, and later was used for intermediate students. The instructor created an assignment with the corresponding review open questions.

The reviewing page included the web form with the list of the review questions that were to be answered according to the reviewed assignment. PRISM showed the improvement in the learning and the students' performances. The system allowed an instructor to investigate the actions of cheating and also showed the un-reviewed assignments or late submission.

PRISM lacked the possibility for the reviewers to be much more competent than the reviewees. And the answers to the web-form was open where the reviewers could type in whatever they needed. It was a good idea to use open and free text to review but it could have been better if they had the reference reviewing document or having a good sample as a reference.

Peer Code Review (PCR) [32] Information System was presented in a programming language course in 2010. 85 fresh students along with few other students from Management Science and Engineering skills were involved in the experiment to test the PCR. Many of the participants were beginners, and hardly few of them had programming experiences. During the semester, the students were required to complete 10 programming assignments.

A survey was conducted to assess the learning outcome. The survey results show that PCR process has enhanced students' abilities in many aspects. Through the questionnaire survey, it was found that the PCR process have achieved good learning outcome as students improved their abilities for programming, conforming to coding standards, punctuality, making suggestions, accepting criticism, and collaborative learning. [32] Although the research is conducted in an organized manner but handling the plagiarism is not taken into consideration.

Another research focusing on the code quality standards at undergraduate level, stated that peer code review helps to improve the quality of coding by the students of all levels. 201 students were evaluated of software engineering course at the Indian Institutes of Information Technology. Even though the peer code review is widely used in industrial and open-source software projects, but rarely taught or practiced at the undergraduate level, especially in software engineering courses. In this study, Bitbucket (in-built code review system with web-based hosting service) was used to extract the peer review comments. It also detected the coding standard or compliance violation and identified defects by the reviewers. The survey concluded that practicing peer code review has incurred a lot of benefits i.e. improvement of coding skills, program comprehension abilities, coding standard compliance as well as communication among peers. [33] This study is limited to Java and Python languages only. Also the severity levels are not clarified, how they are categorized and distributed. Furthermore, the survey results are non-anonymous, which can be biased.

Peer review approach

In order for the peer review to be more useful for either the reviewer or the author, a specific type of peer review must be used to meet the requirements for the assignment being reviewed. There are three used types of peer review, and these are: single blinded review where the names of the reviewers are hidden from the authors but reviewers know the names and identity of the authors. Double blinded is a type of review where both sides are hidden from each other. The third type of peer review is open review where both sides know each other [34].

While using different types and in different reviews, peer review was found to improve teaching and learning where the teaching workload is minimized and the learning improvement is significant. In order for peer review to be successful, it needs an approach which is capable to review any work being reviewed. Here we present an approach used in the PRISM [35] system. They used an approach that allows students to review their peers by filling a free form text, where the reviewing student could write a review based on what assignment he or she is reviewing. They used a form containing two parts; questions and review parts. The review part contained a free space where the reviewing student could write. In this study, a double blinded peer review was used to hide the reviewees from the reviewers and vice-versa.

A-PASS [26], a system for peer assessment, used an approach where the reviews were allowed to be on the top of the original submission, but saved as a review and independent from the original submission. Here they used a canvas technology to achieve their reviewing process. In this study, a double blinded peer review was used to hide the reviewees from the reviewers and vice-versa. This approach showed that it was among the best while assessing mathematics. The researchers here used Fisher-Yates shuffle [36] to generate the reviewer based on the individual student's performance. The assignments were given grades in reviewing but not shown to the author of the assignment, only the performances were shown as either low, medium or high which is more motivating than giving grades to students. The assessment here was done by students giving performance grades to their peers and this can make the authors not trusting the results which could lead to complaints. Here in this project, students are allowed to give comments and feedbacks to their fellows and the reviewer decides on the level of performance the author can belong to. These levels will be averaged and the final level will be obtained which will show the author his/her performance.

After reviewing the approaches used by different researches, we found out that we will use a questionnaire review where the review will be based on specified answers depending on the

assignment provided. This will help in the evaluation of the reviews given by different students. We will focus on the algorithm which will be used to match good peers depending on the intellectual capabilities and also the assessment criteria that can help students in the introductory programming course to learn and acquire skills. This will be feasible as TFLIP evaluates the submitted assignments automatically. According to the peer review papers stated above, we need to take the most powerful and feasible approaches used and use them with the added features of matching the good peers. Also the reference correction will be added for the reviewer to be more confident about the review and there will be no other confusing reasons for not giving trusted reviews.

As peer review was introduced as well as implemented in several tools but it lacked the above mentioned features. We will introduce these features to the TFLIP tool with these added features and hope that it will significantly improve the students' learning. After reviewing the literatures on peer review process, there is a wide gap for knowing what should be assessed (assessment criteria) in the reviewing process. In this case, we investigated the assessment methods available which were done by the researchers and found out some of the assessment frameworks which need to be adjusted in order to be more beneficial and motivating.

In the previous paragraphs, we talked about the peer review in programming in general but for a student to learn programming, s/he starts with the introduction to programming where the basics of programming are taught. After acquiring these basics, a student can be able to program on any level with less difficulties. In order to go further with this introduction to programming learning, there have been researches on this topic and suggested what should be taught to first year students learning computing courses. A multi-national, multi-institutional study was done by researchers on the teaching computing courses in the first year [37]. In their trial survey, they conducted a research to investigate the framework of the steps that can be used to assess computing courses for the first year. The framework included five steps that can be based on while assessing. The steps are: Abstract the problem, generate sub-problems, transform sub-problems into sub-solutions, re-compose the sub-solutions to a working program and lastly evaluate and iterate as shown in the Figure 2 below:

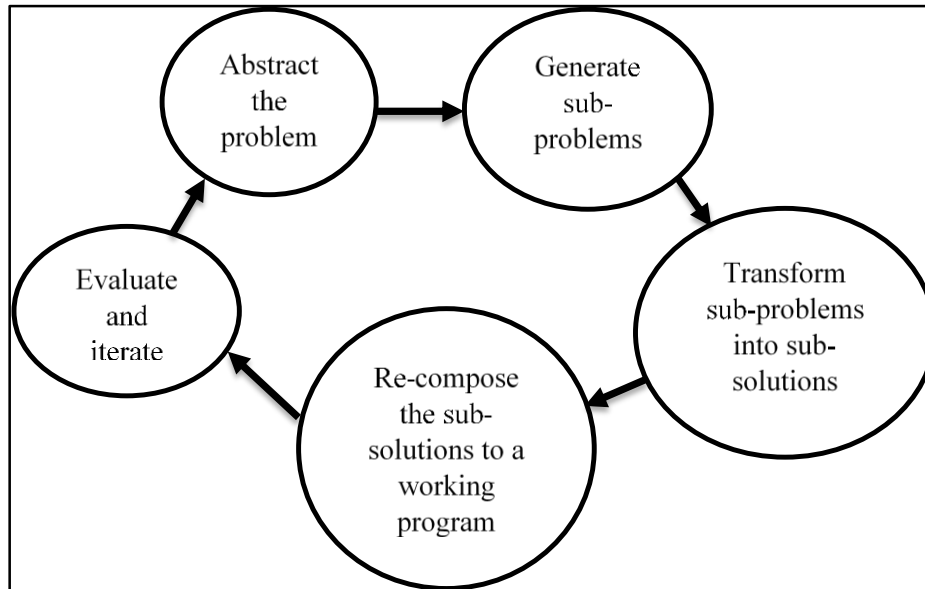


Figure 2: Problem solving steps [37]

This research relates to our work with the difference of our work being concerned with the peer review by the students which we hope gives more motivation in the learning process. The problem solving steps used by the researchers in [37] will be assessed and reviewed based on assessment criteria.

Table 1: Five ways of experiencing learning programming [38]

| Process | Description |
|-------------------|--|
| Following | Learning is experienced as getting through the unit |
| Coding | learning is experienced as learning to code |
| Understanding and | Learning as to write through understanding and integrating |
| Problem solving | Learning to do what it takes to solve the problem |
| Participating or | Discovering what it means to become a programmer |

Secondly, we considered the research on the ways of experiencing the act of learning to programming [38] where the researchers stated five ways of learning programming as shown in the Table 1.

All these ways can help the student to program only if the student has the basics of programming. In this project, we are investigating the ways assessment can be done for the students to get more understanding about programming courses. This is where the question

comes: what are the assessment criteria that can help students understand programming in the introductory course? This question will help us finding the ways assessment criteria can be helpful in the learning of the programming language. Even though it is suggested that learning programming you need to understand the assignment [39], it will also be tested to see the advantages of having the assessment criteria in the learning. We hope that these criteria will improve the learning of the students in the introductory programming courses and students will enroll to advanced levels with the knowledge that can help them develop their careers further.

Assessing the success of an introductory programming course [40] states the investigation done on students enrolled in the first year Information Technology where they were given piece of code as the task to solve and find the output. The study found the task being important to students but with some weaknesses and suggested to improve the ways of assessment. This is where our question still comes as a solution to the suggestion to address the weakness of assessment criteria.

Assessing quantitatively a programming course [41] was carried out and the researchers used measures to assess the students in the programming course. Table 2 shows the measures as follows:

Table 2: Measures to assess students [41]

| Measure | Description |
|----------------|--|
| Effort | The time spent by the student to do the program |
| Defects | Errors in the program |
| Size | Lines of code; counting non-blank and non-comments in the code |
| Productivity | Size divided by effort; and this was an indirect measure |
| Defect density | Defect divided by the size, this was an indirect measure |

These measures were used to motivate and improve students' performances and acquire industrial skills. These measures are well established and we need to use some of them in the development of the framework for the assessment of introductory programming course. They are good in the learning of programming courses but we found them to be more advanced for the students learning the basics of programming courses.

A well introduced and student inspiring is the “problems in learning and teaching programming” [42] where the researchers introduce the study on the learning programming languages as the art of programming that includes knowledge of programming tools and language, problem solving skills and effective strategies for program design and implementation. They mention the common approach to teach programming languages as to first teach the basics of programming languages, guide students towards the whole programming process and later continues to advanced skills. This study suggested that, as learning programming is a difficult task, teachers should design teaching approaches carefully. As it stated the novice programmers by definition as the ones that lack programming skills. The research mentioned also the novice programmers’ errors by Pane and Myers [43] that need to be studied and corrected for improvements.

According to IBM developer team [44], there have been some of the proven criteria that can be used in the assessment of programming languages and these can be improved so that they can be used in reviewing the assignments in T-FLIP.

The criteria stated are as follows:

- 1) Correctness of the code,
- 2) Code length,
- 3) Communication and
- 4) Strategy for coding.

With these criteria, we hope that this project will show the optimal peer review approach in combination with assessment criteria that will be obtained from programming teachers and students learning programming in addition to the related literatures.

Peer review is commonly used in MOOCs as well as other computer supported learning systems. NovoEd is one of the known MOOCs, which uses peer review for their students in order to share and acquire knowledge. They have two techniques used in the peer review process. One is “Evaluation Assigned Randomly” and the second is “Favor those who do more Evaluation” [45]. The first technique is used to assign reviewers randomly while the second technique means that user’s submission will eventually get more evaluations if they have evaluated other submission. This shows that the assigning of reviewers has not been considered in many learning systems and it gives us more courage to do research on the ways peer matching can be used effectively.

After reviewing these both approaches; peer review and the assessment of programming languages especially the introduction to programming languages, we found out that the assessment of introduction to programming needs more clarification and the assessment criteria must be established which will help students learn programming easily. While these criteria will be used in the assessment, they will also be used in the peer review process. This project comes to the main focus of optimizing the peer review in the learning of programming languages specifically in the introduction to programming as a course being taught to the students in the first year information technology at the University of Agder. It will need a survey from both teachers and students to show what they think to be more important in the assessment. Combining these survey results and the literatures, we will form a framework for the assessment criteria and it will be used as a case study for our project.

By not limiting to these main problems, in learning and teaching programming course, they can be addressed by the application of peer review. This will help the students to get more feedbacks from their fellow peers and also they will learn from them. Reviewers will also be learning from what they are reviewing as they are also being reviewed. Students will get more help while using peer review for both sides as reviewer and author.

3 Problem statement and research questions

The techniques of teaching and learning are getting better and better as time continue running. This has made students and teachers for different education levels find the way their duties can be accomplished in an easy and successful ways. As learning is composed of acquiring knowledge from different sources and locations, students need to create their own way of learning from themselves. This makes it easy for them to study their respective courses according to their entire knowledge being shared in order to get more qualified for future competence. Teachers also are struggling to find ways their students can acquire knowledge too. For the teacher to make the students more understood and capable of applying whatever they learn, it is more time consuming and they are overloaded with some other responsibilities and duties.

There have been many ways students can be guided about the studied courses either from parents' or teachers' perspectives. Parents use extra time with their children while helping them about their studies but the parents are also having other responsibilities and the fields of the children may not be matching with their children's fields of study. This causes the parents to give less help about the studies of their children. Also teachers have limited time to spend with students on their studies and tasks needed to be completed by the students in a limited time. Then, students need to get more assistance to make their learning more improving and simplified.

One of the simple way to help students learning is to mark their task solutions and give them feedback. Students get feedback from teachers as either a formative or summative feedback. Although students are getting this feedback, it is not enough to make them convinced about different possibilities that can be used to give a performant result for the task. The students still need to get more sources for the correction of their task so that it can be more helpful for them and also more applicable.

We came to realize that students can get help or support from their fellow students as all of them fall in different performance and competence levels. Students can give as well as receive feedback to/from their colleagues through a process known as peer review process [3]. There have been many ways peer review was conducted in different fields of study and it gave significant improvements to the beneficiaries.

Even though peer review was used and succeeded, there have been more use of random peer matching which was not considered to be more crucial in the peer review process. And this

caused the whole process to give less improvements as expected. Though it is less improving, it helped the needy students who were struggling to get help from nowhere. Peer matching is unexplored field of the peer review field that needs more research on it as it can make the process more helpful.

Particularly, programming language courses are one of these courses that need peer review more than others with the fact that programming is done in different ways but provide the same result. Programming courses students are more specific and need to be guided too according to the programming task they are given to complete. That is the reason for their highly needed peer review process to acquire knowledge from their colleagues on different way the same programming task is being solved by different students. This will help then to reflect to their solution and come up with new or improved solutions and also help them understand programming in their studies hence it will be easy for them in their future programming life.

Having peer review process as a task for student is awesome but also the guidance on the process should be considered crucial too. Programming courses have specific sub topics that needed to be covered. When these sub topics are successfully covered, the student is able to program on her/his level. Also these sub topics need to be verified throughout each program a student submits as a solution to the assignment given by the teacher. These will help to find out whether the student understood and able to use the techniques and methods learnt in the course as well as the correctness of the program code.

As students start from introduction to programming, they need to be guided more in their starting. After they have understood the introductory programming course, they will be more performant in advanced programming courses. For this to be a reality, students need to be provided with a better peer review process with a better peer matching method and in addition, they must be given what to evaluate while reviewing their peers. Students must be given assessment criteria for them to more helpful to their peers and the feedback provided will be reliable depending on the teacher's guidelines.

These assessment criteria up to now have not been explored further. There have been assessment criteria which are more advanced and can be applied to advanced programming courses. Some of these criteria were highlighted in [44], [37] and [3] but all need to be more detailed so as to become more helpful to students enrolled in introductory programming courses. There have been steps to solve programming tasks provided by researchers in [37] too but were found to be more helpful for the advanced level students or programmers.

The purpose of this study is to determine an optimal peer matching algorithm which will be used to group and match the peers for assessment to enhance the learning of programming language abilities for the students using peer review process. This project is studying the state-of-the-art to discover related to successful peer review methods and approaches. We will use these approaches to make the improved peer review which will be implanted according to T-FLIP learning management system. Following are the considered hypotheses and research questions for this project to be effective:

Hypotheses

1. Using optimal algorithm for matching peers is more beneficial than random peering approach in the peer review process.
2. Having well established assessment criteria related to the task in the introduction to programming course helps the review process work and the assessment of programming course.
3. Applying the peer matching algorithm to T-FLIP improves the usefulness of the whole system.

Research questions

- Which is the optimal algorithm/approach that can be used to match peers that will be used in the peer review process?
- What are the assessment criteria for reviewing the programming assignments which can be adapted in peer review for the introduction to programming course?
- How can the approach concept be adapted to TFLIP features and data structure to improve the system usefulness?

The selected, optimal peer review approach will be tested and evaluated with the proof – of – concept implementation in a peer review component that will be developed under T-FLIP data structure and features. This must comply with the assignment task terms in the tool and also be able used by the students who submitted their solutions to the assignment. A prototype will be developed for the evaluation of the stated research questions where and must follow T-FLIP data structure.

4. Methodology

This chapter highlights all the means of research being conducted. The research undertakes both qualitative and quantitative methods in the form of online surveys and individual depth interviews. The research methodology explains what and how the research methods are used for data collection and its further analysis later.

4.1 Data collection on assessment of programming courses

This project focuses both on learning and involving computers within educational spectrum. It is therefore important to mention that the methods and concepts are inspired and taken from human computer interaction (HCI) paradigm.

The main participants for this research were 4 teachers of ICT department at University of Agder and 50 students studying computer programming course at Bachelors level. The students learning programming language course ranging in age from 18 to 30, speaking different languages and having different cultural backgrounds. Whereas, the teachers teach different programming courses at the university of Agder and age ranging from 35 to 58. Teachers and students are the core users of the peer review system, therefore it is important to consider their requirements to make the system more efficient and effective.

4.1.1 Interviews with teachers

First of all, it is imperative to get the overall feedback of teachers about the peer review process. Secondly, how the peer review can help the students to improve their learning of programming language courses? What assessment criteria are to be used to assess the students' performance? To get answers of these questions, we will use the interviewing technique. We will consider different teachers from ICT department at the University of Agder for interview. An appointment to conduct the interview will be taken by contacting them via email. The interviews of the teachers will be recorded using a voice recorder and their consent will be taken before the beginning of the interview.

During the interviews, semi structured technique will be used for individual depth interviews. The interviewees will be asked a series of questions prepared beforehand and a summary of these interviews is listed under appendices. The desired outcome of the teachers' interviews during this exploratory study is to find out answers relevant to the research questions. the development of the peer review system will be partly based on the responses from teachers. The second part will be the survey from students to get their perspectives.

4.1.2 Survey on students' perspectives

Surveys are extremely helpful ways to reach the target audience in different places where it is difficult to reach physically. It provides fast response to collect real time results and basis for analysis. Online surveys provide number of benefits over manual surveys. First of all, they are inexpensive to conduct, no paper and printing cost to bear. The online surveys also provide automatic summarization which makes this more accurate over manual surveys. It is also observed that the online surveys are easy to use for most computer or tablet users. The surveys can be designed colourful and made attractive for the participants to keep them focused.

In order to gather information from students regarding assessment criteria to assess the performance, an online survey will be tailored as a basis of hypotheses. This method of conducting the survey is used as a descriptive method to collect the data from a large group of 120 students. These students are enrolled at bachelor's level at University of Agder. It also provides required information which helps in decision making.

For this purpose, an online survey form will be distributed among the students to get the students' opinion about peer review and assessment criteria. The web link will be provided to the teacher for the distribution among students through their emails. This online survey is supposed to provide two major benefits:

1. Students were free to give their input regarding peer review and assessment criteria.
2. Saving time to gather information from a larger group.

Interviewing one to one could be time consuming and therefore not an effective method to be used when the participants are a large group of students.

4.2 Data analysis

Data analysis is a process to inspect, clean, transform and model data to attain meaningful and useful information to derive conclusions which facilitates in decision making. Analysis of data helps us in many ways and plays an important role, the data is analysed and on the basis of important decisions to be made. Analysis of the data that will gathered include both qualitative and quantitative data that are expected to be obtained from the core users of the system.

4.3 Human-centred design

According to ISO 9241-210(2010), the human-centered design is an approach to interactive systems development that aims to make systems usable and useful by focusing on the users,

their needs and requirements, and by applying human factors, and usability knowledge and techniques [46]

The term human centered design and user centered design can be used interchangeably. It's a research and design methodology where the process starts with the people for whom the system is designed for and leads to final solution which meets their requirements. It revolves around the needs and wants of the end user and the solution which is provided as a result. This solution is designed according to their needs rather imposing the solution to the end users and letting them use it. Therefore, it is necessary to test, evaluate and iterate the test phase to improve the solution which matches the exact requirements of the end users.

The design of the peer system in this project is based on the human centered design process as per ISO 9241-210 shown in the Figure 3.

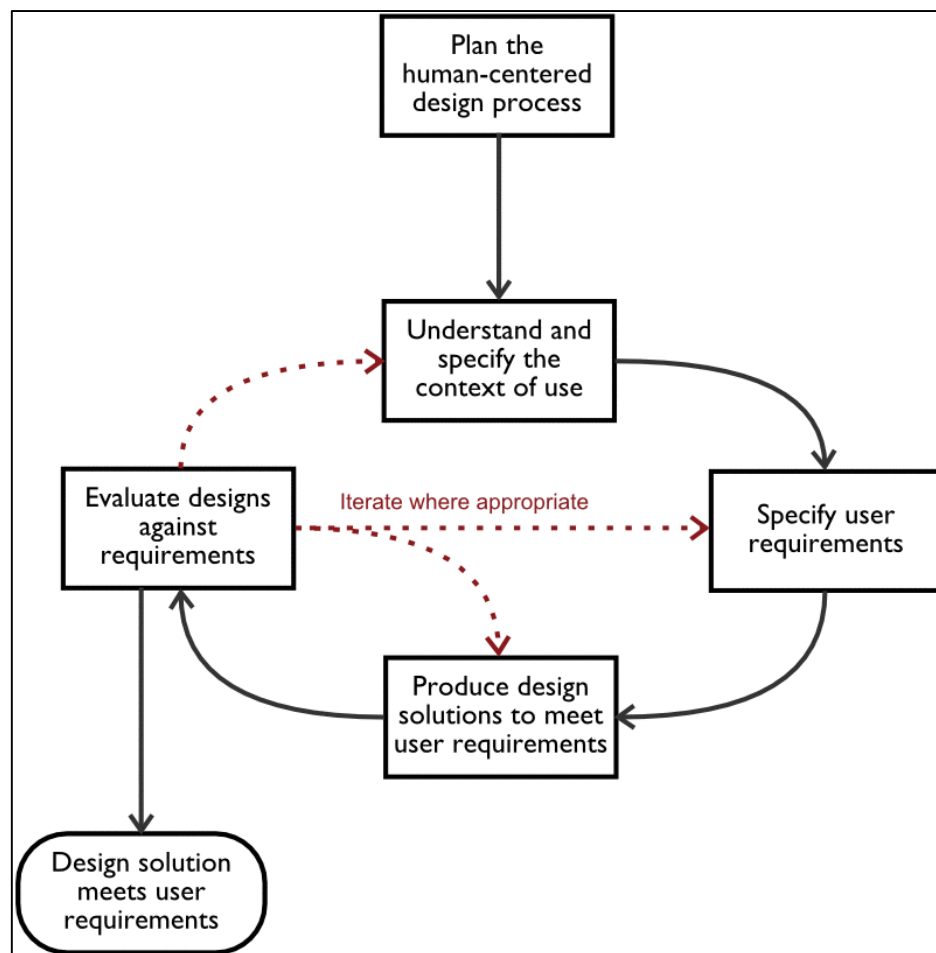


Figure 3: Human-centred design process [46]

The human centred design process starts by planning the design process that will lead to the completed and functional system or product. This stage considers the time limits for each process and who will be concerned in the process. User requirements need to be specified

before the system is developed. These requirements must be obtained from the system users. After the establishment of the requirements, the development phase starts. The development is evaluated based on users' needs and the specified requirements. Depending on the users, there can be new requirements to be added to the system. The system is developed based on the evaluation by the users where they specify new requirements and be fixed. The system is iteratively evaluated to meet user requirements.

4.4 User testing and evaluation

User testing refers to evaluating a product or service which is to be tested by the users of the system. Mostly, the first iteration of user testing requires more time for the designers to setup the test and also for the users to understand the system. It is where the core users and the administrator or designer sit together to operate and to observe. Mostly the user testing is conducted with single users. It is important to identify errors and mistakes to define new requirements for the system to perform as expected by the users.

In user testing, problems can be found on the basis of error/failure rates and have to be analyzed closely to find the root for every failure. From the user's perspective, getting started with a new product is difficult. At this point, we want users to use the system and express the difficulties they face by criticizing it. Therefore, it is highly important to iterate the design phase cycle to test, detect problems, measure, and then redesign. These steps should be repeated until the design solution meets the user requirements. The test is conducted under controlled conditions. From ethical aspect, it is recommended to disclose the research purpose and get the consent form signed by the potential participants.

5 Requirements engineering

The requirements engineering is a detailed process which analyses, identifies, documents, specifies and maintains requirements. It also specifies the purpose of software, its usage and under what environment it is being developed. Requirements can be qualitative, quantitative and organizational. It also explains the functionality and the expected outcome of the product being developed. The requirement engineering process saves the time and effort which is required by the software developers in order to complete the task as well as minimize the development costs. All the aspects, for example: how the product will interact with hardware and other systems and with the users are discussed. In order to obtain the requirements for the peer review system, we conducted interviews and survey from teachers and students respectively.

5.1 Summary for teachers' interview

Interviews were conducted with teachers to find out important requirements which can be met by the system to be beneficial for both students and teachers. Four teachers of computer programming courses were interviewed and a summary of these interviews' outcomes are described as under:

Teachers were asked if they find the peer review system to be beneficial for both students and teachers. They all agreed on the benefits of the system and said that students will benefit from the system by learning from their fellows and be able to correct their solutions to programming problems. Also teachers' overload will be minimized.

Teachers were asked if they should be involved in the peer review process. They responded by confirming that they should be able to view the reviewed assignments as well as being able to review them if requested by students. They also said that if teachers are not included in the process, students may not take the process seriously. Three out of four teachers agreed that the peer review process should be anonymous.

They were asked about the reviewers' distribution and while they are preparing the review process. They responded that; students have different capabilities on which the distribution can be based on. Students must be grouped in three performance groups or levels. Each student must be assigned to at least three reviewers, one from each level. Teachers said that each student should receive three to five review per assignment and all should be considered the same way.

When teachers were asked about the assessment criteria that should be followed during the peer review process, they said that all the topics covered by that assignment should be used. Assessment criteria should have weight according to their importance in the solution. They should be provided by the teacher and saved with the assignment problem. Assessment criteria must not be available to other assignment other than the one they are assigned to. They must only be retrieved during the review process of the corresponding assignment.

Next, they were asked about the feedback type which can be used in the peer review for programming language. They responded that as the students are not allowed to give grades, they can only give comments which corresponds to the assessment criteria.

5.2 Students' survey data analysis

Out of 120 students only 21 responded to the online survey. The questionnaire had 20 questions in total, 2 open ended and 18 close ended questions to answer regarding peer review. Each close ended question asked, had minimum two and maximum five options to select one from. Students could give their opinions by selecting the appropriate option from the online survey.

After compiling the data, we gathered from teachers and students in the form of interviews and surveys respectively, the process of analysing the data begins. At this step, we try to analyse the results from the data provided by the teachers during individual interviews, as well as from the feedback gathered from the students via online survey.

To assess the performance of the students taking programming language courses, it is critical to first establish an assessment criterion. The assessment criteria will later help in the design phase of peer matching algorithm. After forming the assessment criteria and going through the peer review process, the performance level of students is obtained.

We will now take a look at the survey questions and analyse the results. Some questions asked are explanatory where students have different opinions whereas others are presented on the pie. The pie charts are used to illustrate the percentage and proportion of the data collected.

In the early questions asked from the students during the online survey, they were asked if they have ever used the peer review process. In Figure 4, at first glance it is clear that the majority (62%) of the students answered that they have used the peer review process before and they are aware of it, whereas 38% did not use it previously.

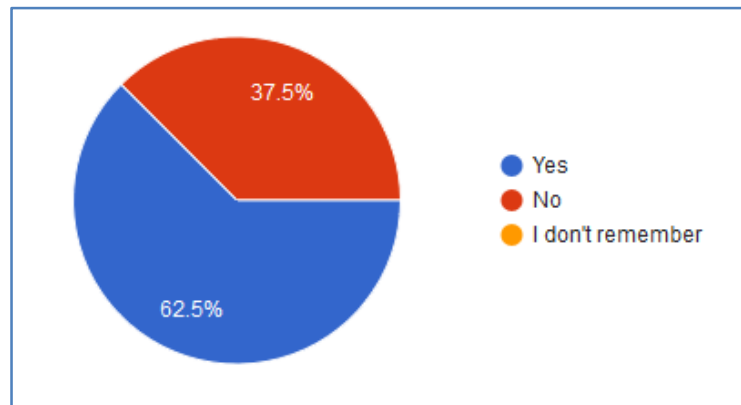


Figure 4: Previous experience of peer review

In the next question, the students were asked if they would like to know about how to review peers? Once again the majority (66%) of the students were not interested to learn how to review peers but 33% are willing to learn about the process of how to review peers.

Looking at the Figure 5 **Error! Reference source not found.**, at first glance it is clear that the main influencing proportion of the students who strongly agree that the peer review can be beneficial for students. Supporting to this, 29% of students chose to agree to the benefits which students may get to improve their learning ability. Making it nearly 80% who agree with the benefits peer review may bring in their learning. Whereas 16% somewhat agree and the rest neither agree nor disagree.

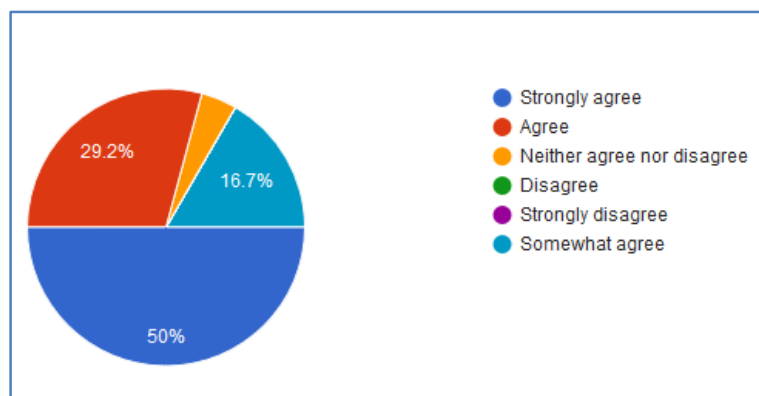


Figure 5: Benefits for students from peer review system

This question is open ended, where students are allowed to record their feedback whether they would like the teacher to be involved in the peer review or not. Most of the students agreed that the teacher should be involved in the process and should review the submissions and give feedback.

Figure 6 shows a significant number of students (87%) would like the review process to be anonymous, where the identity of the reviewer is not visible to the reviewee. Whereas a minority of just 12.5% students are interested to make the review process transparent, so that

the identity of the reviewer is known to the reviewee and the reviewee can actually see who has reviewed his or her submission.

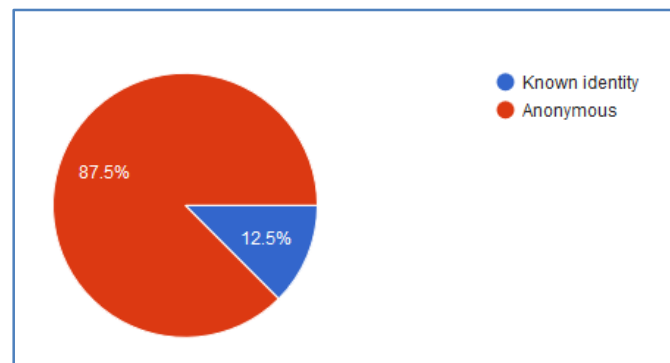


Figure 6: Peer review by anonymous or known identity

In Figure 7, more than half of the participants (54.2%) of the survey are interested to have a mix of Low, Medium and High performance levels to be assigned for review to one assignment. A minority of 16.7% wants the reviews to be made from the same performance level whereas less than a third is interested that the reviewers should be assigned randomly.

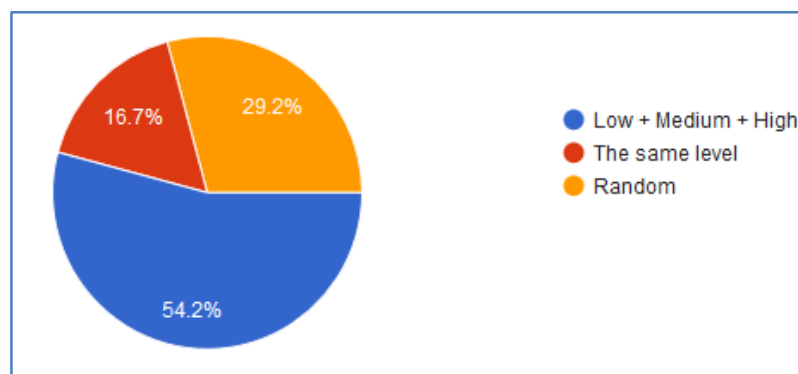


Figure 7: Students' grouping

In the next question asked during the survey regarding which type of feedback the students would consider the best, should it be comments, grades or both. Figure 8 shows that 83% of students expressed that both grades and comments is the best way of getting feedback. Only 16.7% want only the comments and not the grades as feedback.

Students were asked how many number of reviews per assignment they consider to be sufficient and reliable. As shown in Figure 9, seven in ten (70%) consider 3 reviews per assignment would be reliable enough. Whereas 16.7% of the students think that 2 reviews per assignment are sufficient. Surprisingly just over 8% students would want only 1 review per assignment whereas the minority of 4.2% would like to have 5 reviews per assignment.

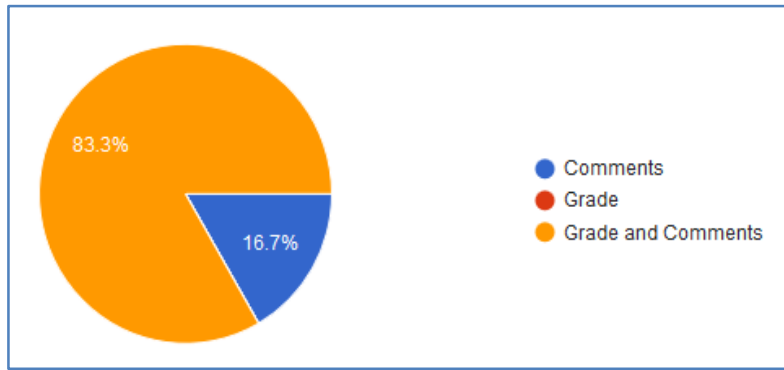


Figure 8: Feedback consideration

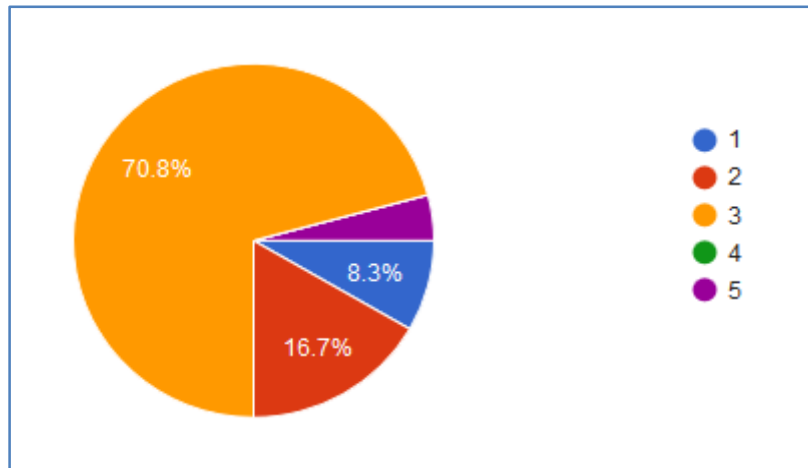


Figure 9: Reviews per assignment

In the programming code specific questions, the feedback collected from the students.

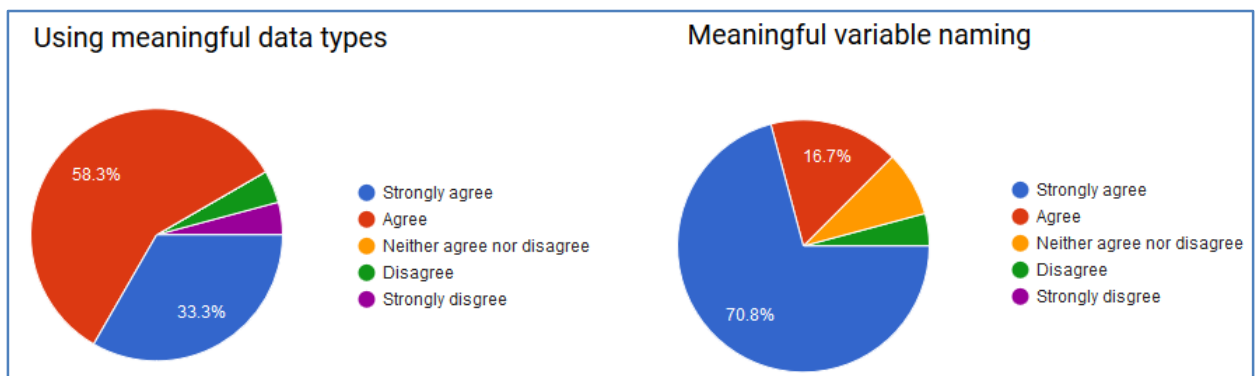


Figure 10: Using meaningful data types and variable naming

To use meaningful variable naming while writing program code, more than 70% students agree to this fact and just above 16% support by agreeing that the variables should be named meaningful and not just like xyz or abc.

In Figure 11, almost 90% students agree that the syntax should be evaluated and must be reviewed, as without correct syntax the code may not work and the whole program does not show the output at all.

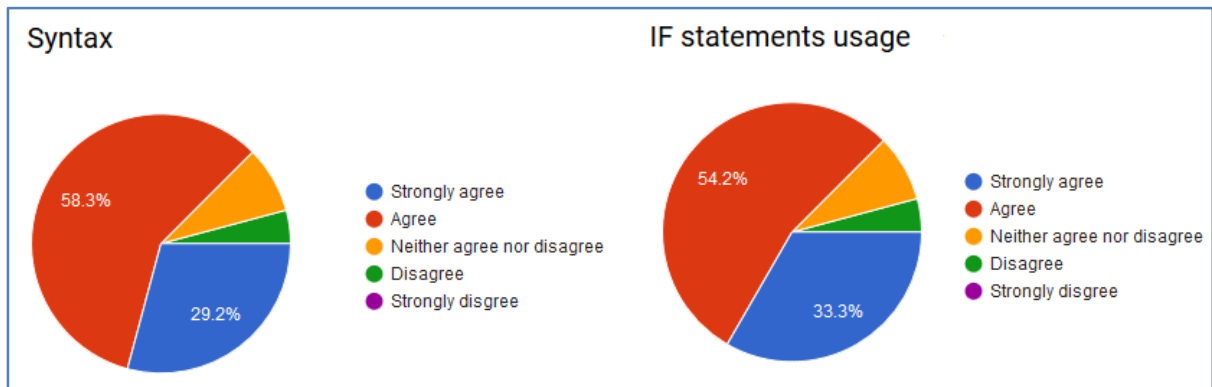


Figure 11: Correct use of syntax and usage of If statements

On the right side of Figure 11, more than three quarters agree that If statement should be reviewed and checked whereas the minority of one quarter either disagree or fall in neither agree nor disagree for the favour of If statements usage.

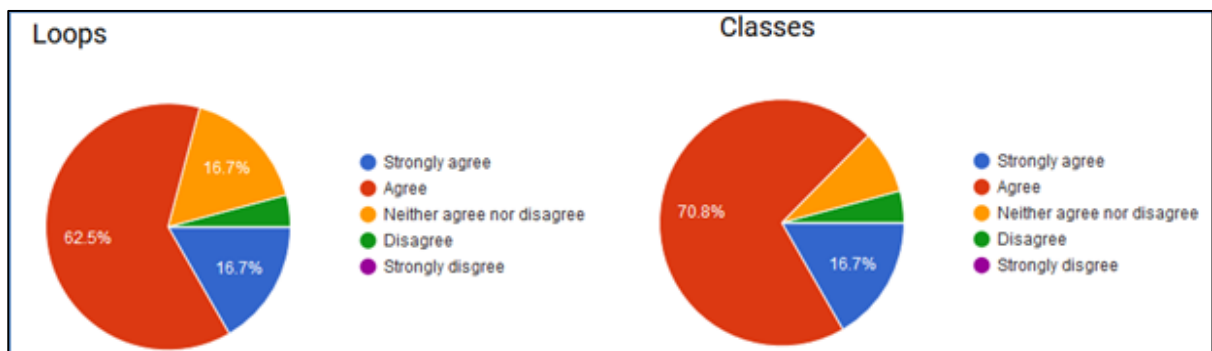


Figure 12: Usage of loops and classes

In Figure 12, while the students are asked about their feedback regarding loops, over 3 quarters think that it is important factor to assess and should be in the assessment criteria, whereas a minority of 22% students either disagree or neither agree nor disagree.

On the right side of Figure 12, over 86% agree to assess the use of classes in the programming is an important factor to be assessed and reviewed. Whereas less than 14 percent do not agree to this.

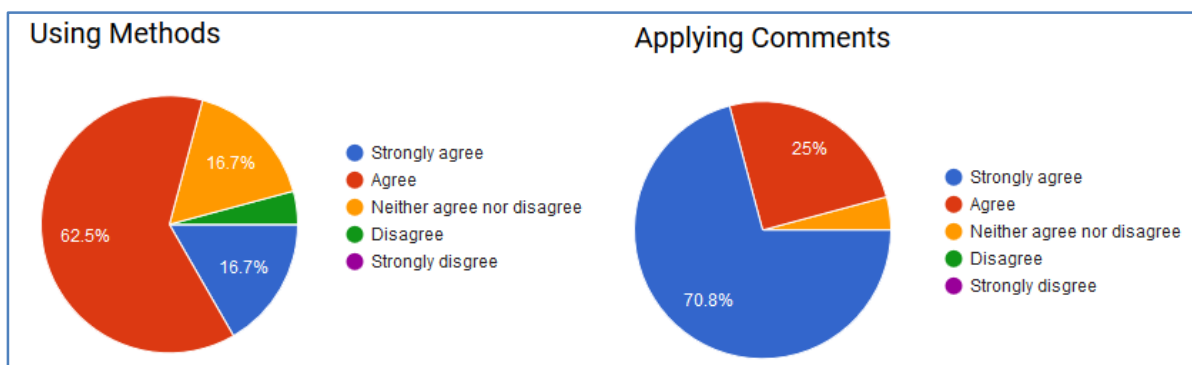


Figure 13: Using methods and applying comments

Error! Reference source not found. shows that the use of Methods in the programming code and assignments should be assessed. Over three quarter of students agree to this whereas nearly 20% of students either disagree or say that they neither agree nor disagree. On the right side, a major portion (95%) of the students agree that comments should be applied along with the program code, to make the code more understandable by the readers, whereas less than 5% of students neither agree nor disagree.

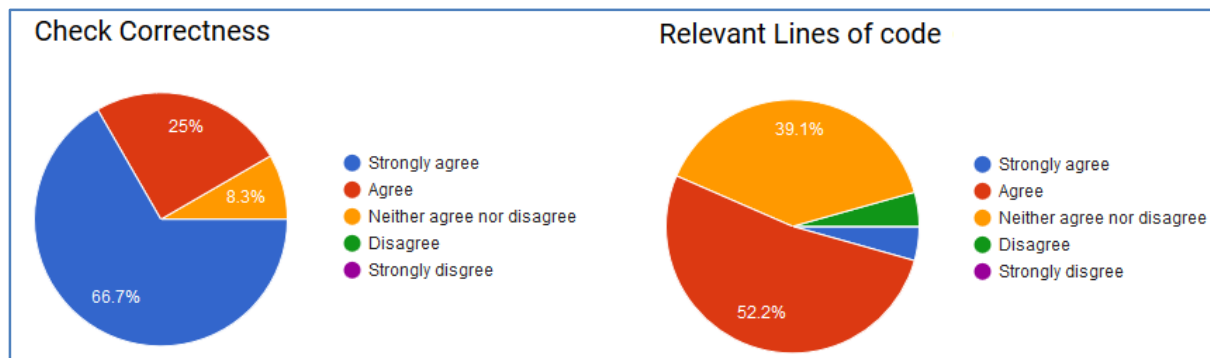


Figure 14: Check correctness and relevant lines of code

In the next feedback, as shown in the Figure 14, over 91% of students agree that the correctness of the program should be checked and evaluated whereas just above 8% of students neither agree nor disagree to this factor. On the right side of Figure 14, when students are asked to give their opinion whether relevant lines of code are important to be assessed or not, over 91% agreed whereas the rest either disagree or neither agree nor disagree.

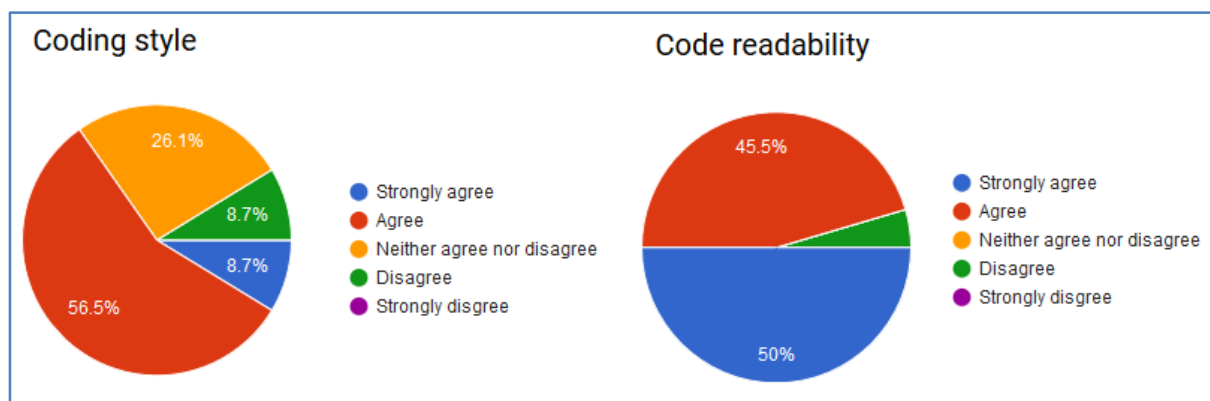


Figure 15: Coding style and code readability

In the Figure 15, over 82% believe that the coding style should be assessed during the peer review. Only 8% neither agree nor disagree and another 8% of students disagree to this factor. On the right side of the Figure 15, a very large number of students just above 95% are in favour to include code readability to be assessed whereas less than 5% disagree to the majority.

Based on the data gathered from the students via online web survey, it makes it quite clear that what factors influence the assessment criteria and what must be included for the assessment of students' work done as assignments. This collection of data will now help us to prepare assessment criteria and it will further be given weightage so that peer matching should be made effective which will benefit the students.

5.3 Requirements

Requirements engineering is one of the most critical part in the development of any system which will be used to solve any given task. For us to specify and organize the requirements for this project, we used a Volere requirements specifications template [47]. This template shows different critical parts of the project including the tasks to be done using the product (requirements), developers and all stakeholders. In addition, it shows the properties of the environment where the product can survive and benefits to that environment.

The purpose of the product

Peer review is a process where students evaluate their fellows and give them either feedbacks in form of grades and/or comments [48]. This project aims at giving comments feedbacks to the students learning programming languages and also grouping the students in accordance with their performance levels according to the users' requirements. The levels to be used here are high, medium and low level. These levels will be based on the students' performance in the current assignment being reviewed and also if possible, the previous assignments done by the student. The main purpose of the peer review in this project is to optimize the learning of programming courses where students will learn from their fellows' works and also from the comments they receive.

Stakeholders

This product will not be sold to any institution; it will be used by the University of Agder (UiA) for teaching and learning purpose. This is in accordance with T-FLIP tool at UiA, where it will be integrated to support peer review as T-FLIP must have this feature so that it can be helpful for the students to review their fellows' assignments.

In this case, UiA will be the client as well as the customer for the product because this is a thesis that must be done for the completion of the degree and also to help the university accomplish its target of having a fully functioning learning system with peer review component.

For this project to successfully completed, all stakeholders will be involved in the development for it to fulfil the requirements needs all users. The identified core stakeholders include the teachers for programming languages courses, developers for the product and the students using the product. They will be used to generate the assessment criteria suggestions which will help in the development of the assessment framework for the programming courses.

Teachers will be requested to give the assignments and also their corrections (marking scheme) for comparison reasons. After the students have submitted solutions to the assignment, they will be allowed to perform review process for the evaluated assignment. After all, teachers will have an option to review any student's work as it could be requested by a student. Teacher is allowed to view the given reviews to all students.

Users of the product

Users of peer review in this project will be the teachers who teach programming language courses and the students learning programming. There will be the support team after the completion of the product so that the system will be maintained and updated as it will be required. The role of the teacher will be the preparation and submission of both assignment tasks, marking scheme and the assessment criteria. Students will use the product to deliver the solutions to assignments, to be part of a peer review process and they will receive and learn from the received reviews.

Project constraints

This project must be able to accept the assignments given by the corresponding teacher with its corrections. Students who are registered for the course must be able to do peer review process. The final product must be available online for both wireless and wired networks connected to internet.

Functional requirements

The scope of the work

The work for this project will be concerned with the development of an algorithm that will be used to group peers according to students' performances. The reviewing process interfaces and the assessment criteria will be developed which will be used in the peer review process. The main focus of the work presented here is developing the idea and the structure of the peer review process which will be implemented in the T-FLIP tool after completion of the product. Only the programming language courses assignments will be reviewed using the project after

completion. The product will help students receive feedbacks (reviews) from their fellow students. In order to fulfil these tasks, we need to investigate the performance levels of the students so that they will be grouped according to the performance. Also the programming language assignment assessment criteria must be investigated so that they will help in the review process.

The scope of the product

The product will be used by teachers or instructors that are concerned with programming language courses and the students learning the courses according to the project description. The main focus of the project is the matching of the peers and the reviewing process. Here below is the use case diagram for the project. The use case diagram shows the tasks being done by the system from the teacher preparing the assignment, submits the assignment to the system, also prepares and submits the marking scheme for the assignment to the system. The marking scheme will be compared to the students' assignment correction/solution for the evaluation purpose. Teachers also prepare the review processes' assessment criteria and they are attached to the corresponding assignment. Students open the assignment, solve it and submit the corrections/solution. The student runs the auto-evaluation to check the correctness of the assignment and also receives an automatic feedback if it is a pass or a fail. The students are grouped as peers to review each other's assignments. After the review process, all students receive the reviews for their works. In this step, students have to read the comments from their fellow students and make possible corrections for their works. The use case diagram below shows the process of the system after the user has logged in to the system as a registered user. Red ovals represent existing features while the blue ones represent new features.

In this case diagram shown in Figure 16, there are two features shown by a bi-directional arrow, one showing the automatic evaluation and another for the peer grouping. Peer grouping will be the most important in our project as well as peer review itself as a process. This is because automatic evaluation is already in existence with T-FLIP tool.

Functional and Data Requirements

Functional requirements are the fundamental subject matter of the system and are measured by concrete means like data values, decision making logic and algorithms [14].

This project is based on the requirements specified by the users and the literatures about the existing systems which have related intentions. The first and foremost is the grouping of the

peers. “Peers in this system will be grouped according to their performance levels where three levels will be grouped together to form a reviewing group” said a programming teacher during the interview. These performance levels are high, medium and low. For the group to be relevant, it will have students from each level so that the reviews will contain different expectations or logical thinking.

A student must receive at least three reviews from all the groups, and must also review three students. In addition, the teacher will be required to make a review whenever requested by a student through a complaint. All the reviews must be considered the same way and no review should be ignored.

The review process will be done in accordance with the assessment criteria set by the teacher in the preparation of the assignment. The teacher will also provide concrete answers that will be used to give feedbacks to the work being reviewed. “These criteria must be dependent on the assignment tasks” said a programming teacher, and they will be used by the students in the review process. This will help the reviewers to review assignments with specific feedbacks set by the teacher and also they will be more effective in the management process by the teacher after the process.

A peer grouping algorithm groups the students (peers) according to their performances. These peers are able to review each other’s work with the help of assessment criteria. After the process is complete, the reviews are sent to the owner of the work being reviewed. Also reviews can be seen by the teacher responsible for the course with the assignment.

Here are the requirements for the system to be functional and helpful to either the teachers or the students leaning programming courses.

1. The system should accept the assignments submitted by the programming language teachers. The document example is shown in the Figure 38. The system accepts pdf, doc/x, txt file formats.
2. An assignment should contain the marking scheme as a reference solution for comparison with the students’ corrections, an example is shown in Figure 39. The solution file must be uploaded to the system and must be of either txt, json or java file format.
3. Assessment criteria must be stored to the according assignment and be able to be used in the reviewing process. They should not be viewable from other assignments.

4. Teachers should be able to test whether the assignment is well set by evaluating their own corrections to the assignment. The system must allow the teacher to resubmit the assignment questions or solution when necessary.
5. Students must be able to view all assignment assigned to the course on which they are registered, and must be able to download the assignments.
6. Students must be able to submit their correction to the assignment task.
7. Students must be able to run the automatic evaluation for the submitted work and get the evaluation feedback as pass or fail.
8. Students who get fail as the feedback should be able to re-submit the correction and run the evaluation again.
9. All students' performances must be recorded and able to be retrieved when needed. These performances must be quantitative for them to be used in the sorting and comparison of students.
10. A grouping algorithm must group the peers (reviewers) according to their performances on the current assignment. Also should consider the previous assignment performances.
11. Reviewers must be able to review a number of works submitted by other students. According to the programming teachers, it is recommended to have 3 reviews per one student.
12. The teacher or instructor should be able to review any assignment when required. The system must allow teacher to review any student when needed.
13. The reviewed student should receive all the reviews and be able to read them all.

In the above list of requirements, from requirement 1 to 8 will not be focused in this project. Only requirements from 9 to 13 will be the focus work of this project according to the users' requirements in the conducted data collection.

Use case diagram

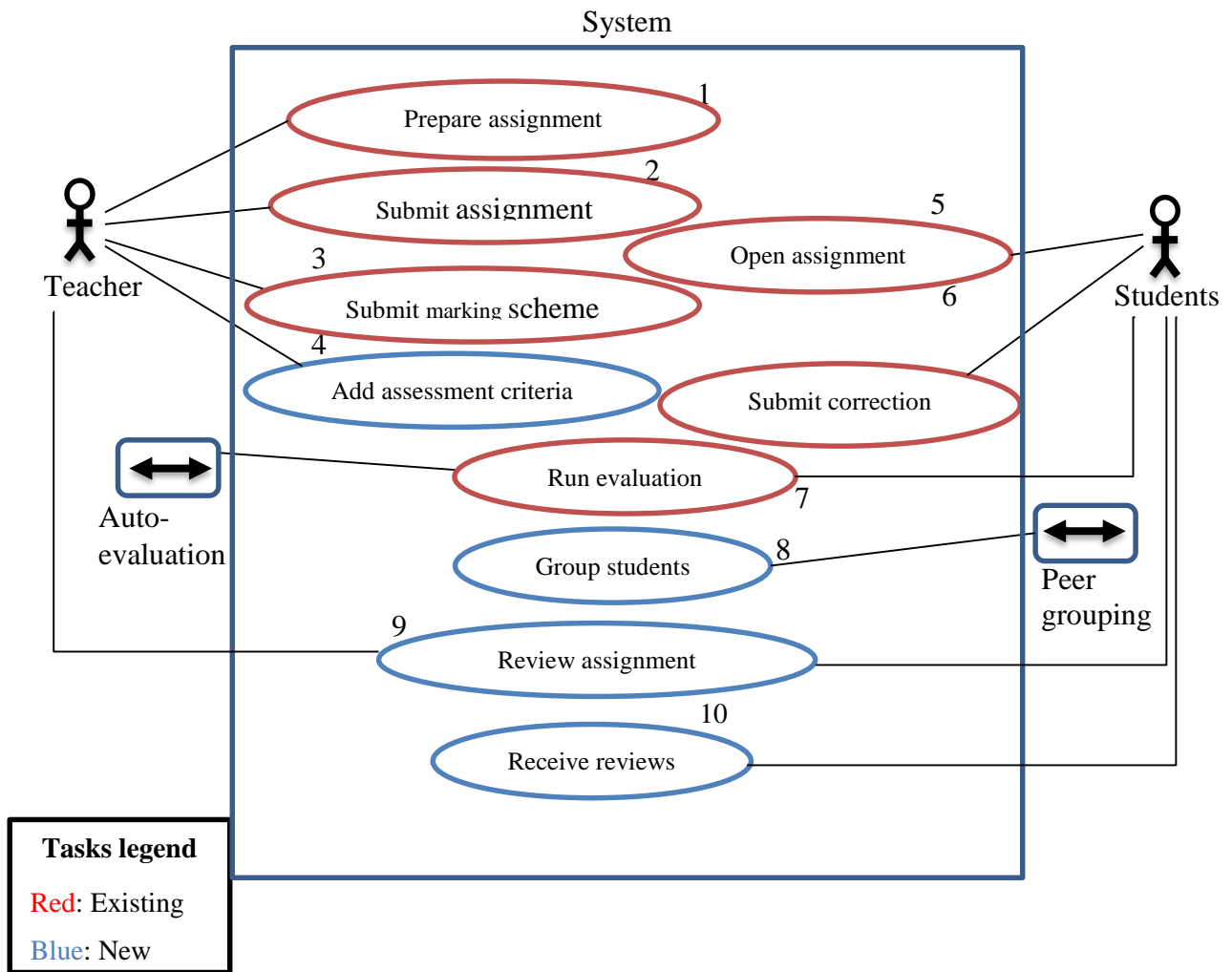


Figure 16: Use case diagram according to Volere template [47]

Non Functional Requirements

The non-functional requirements define the overall qualities and attributes of the system. These requirements place restrictions on the product being developed, the development process, and specify external constraints that the product must meet, for example: safety, security, usability, reliability and performance requirements. Project management issues (costs, time, schedule, etc.) are often considered as non-functional requirements as well.

Look and Feel Requirements

This section highlights the requirements of the interface layout. Depending on the user requirement regarding corporate branding, style, color and the degree of interaction, etc. are considered while designing the interface. The product will be available online on UIA's server with high speed Internet. The physical server to be used will be determined by the IT

department. The speed of the reviewer's connection will depend on the connectivity and hardware used rather than characteristics of this system.

The web based application will run on the reviewer's PC and will require a web browser which is mostly installed and available in all computers. It will be available for both Windows and Mac operating systems.

Usability Requirements

The usability requirements cover efficiency of use, ease to remember, error rates and overall satisfaction to use the product. The application will provide a simple yet easy to use interface for the peers. The users will be able to download the work done by other peers as well as the criteria provided by the teacher. Similarly, the feedback and reviews will be uploaded by the same application. Help will be available as required to avoid the chances of mishandling and to make the tasks easy to understand. Also, the users will be provided with the adequate training to use it efficiently. The application will initially be available in English language with a further possibility of Norwegian language to be implemented in the future.

Performance Requirements

Performance requirements specify the amount of time which is available to complete specified tasks. These tasks often refer to response times and to the product's ability to fit into the intended environment considering the safety concerns. Furthermore, it is important to quantify that up to what level certain accuracy in the produced results is required. It is also important to define the necessary reliability and availability of the product. According to the requirements specified by the users, the initial interface will be simple and easy to understand, so that the load time and maximum response time is reduced to minimum.

The task to download and upload the review text or document would be possible for a maximum time of 10 seconds. To make the system more reliable the product shall achieve 99% up time. It will be available to its users 24 hours depending upon the task submission deadline. In case of any disaster or abnormal happening, the product may stop to respond. In such circumstances, the process to download and/or upload the task or review may need to be done once again. As this application is limited only to programming language courses which are offered at the university, no limit has been set for the users/active students to connect to the system whereas the system will be capable to cater a volume of 200 students at one instance of time.

Operational Requirements

These requirements specify the physical environment in which the application will operate in its intended environment. Also, if the product needs access or work with other partner applications. In such cases, the application must interface with the partner application(s). This particular application can easily be accessed by students and teachers from anywhere, as long as the person has access to the system and have rights to operate it. A standard desktop computer or laptop and even mobile phone can be used having a web browser to load and use the application. The application must be accessed over internet using web browser. For further technical requirements refer to user manual, online help or the support team can be contacted for assistance.

Maintainability and Portability Requirements

Maintainability and portability requirements are based on the time which is required to make specific change or update the product. It is also to aware the users and stakeholders about the maintenance of the product. Our application ensures that the updated reviews will be made available to view as soon as they are uploaded by the other peers for that particular peer. To make it possible, an email notification will be sent to the relevant user regarding the update. Similarly, if a peer has a pending review, will be notified about the task to be completed.

The maintenance of the system is scheduled once in a semester time frame and the users will be informed about the maintenance period and downtime (if applicable). Every registered user will be able to access the application via internet. The application is expected to run under Windows, Linux and Mac operating systems. It will primarily be designed for desktop and laptop computers, but will be accessible on tablet PCs and mobile phones as well.

Security Requirements

Security requirements are one of the most important requirements to focus on. It is important to specify that who has authorized access to the product, and under what circumstances that access is granted, and to what parts of the product access will be allowed. In this case, teachers and students are the main users of the application. They users will be provided with different access rights, for example Create, Read, Write, Delete and Print. Table 3 shows the access rights is given below:

Table 3: Roles of system users

| User / Rights | Create | Read | Write | Modify | Delete | Print |
|----------------|--------|------|-------|--------|--------|-------|
| Admin | Yes | Yes | Yes | Yes | Yes | Yes |
| Teacher | Yes | Yes | Yes | No | Yes | Yes |
| Student | No | Yes | Yes | No | No | No |

Furthermore, the support team from the IT department, programmers, program coordinator will also have specific rights to make changes and to allow or disallow rights to various users. To ensure the confidentiality and security of this application, only direct managers will be granted full control rights.

The Role Based Access Control (RBAC) states that permissions are associated with roles, and users are assigned to appropriate roles. This greatly simplifies management of permissions. Roles are created for different job functions in an organization and users are assigned roles based on their responsibilities. Users can easily be reassigned from one role to another. Roles can be granted new permissions as new applications and systems are incorporated, and similarly the permissions can be revoked from roles as needed. [49]

We use Role Based Access Control means to implement the security where the decisions to access the application are based on the roles of the users. The users of the system are assigned roles as Student, Teacher or Administrator. Based on these roles the users will be able to get access to the relevant functionalities of the system.

This application will be embedded into the T-Flip system, and for this reason the security is crucial so that any unauthorized users should not get access to the system. The students and teachers will be notified as and when a feedback, review or task is available. Similarly, whenever the update is available the relevant user will receive an alert by email.

5.4 Peer matching algorithm

In order to have all students assigned to reviewers in the same way, we designed an algorithm that will be used to match the peers. It will be able to fetch the students' previous performances from the stored records and use them to find out the best students to review others according to their performances. The performance to be used will be the total points obtained from the assessment criteria as were reviewed by the assigned reviewers and the combination of individual assessment criteria points. The algorithm sorts the students according to their performances and they are classified in levels according to the combined criteria.

Combining all these criteria is purposely for making the process personalized so that students will be assigned to reviewers of different performances and different from what they possess either in total or in individual criteria. The Figure 17 shows the summary of the peer matching algorithm. The algorithm will start by sorting the students according to their performances in ascending order. This will help the students to be categorized into three performance level which are Low, Medium and High.

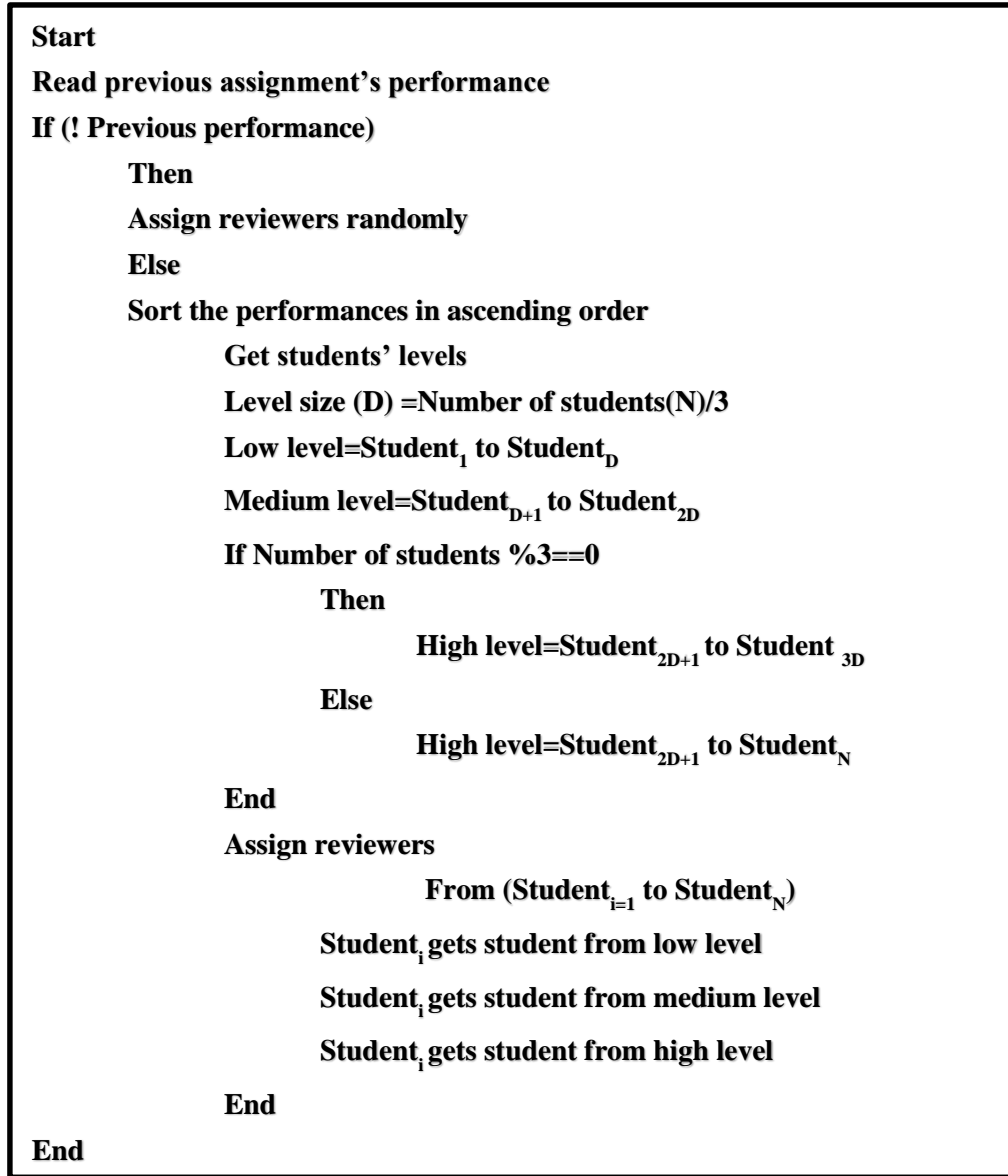


Figure 17: Peer matching algorithm

The algorithm will use a modified Random Subproducts algorithm [50] to assign the reviewers to all students from the three level after they are classified in level groups. One student will be assigned with three reviewers one from each level and also one student will be given to review three fellow students. Random Subproducts algorithm will be used to randomly select a student

from each level so that three students will be selected from three level and they will be given to review one student from any level as presented in Figure 17.

In Figure 18, there are three students' levels: Low level, Medium level and High level. The low level is composed of the students with the low performance from the sorted performances. The students are sorted in ascending order based on their performances. The medium level is the middle group of students according to their performances while the high level is composed of those students that scored the best in the previous assignment.

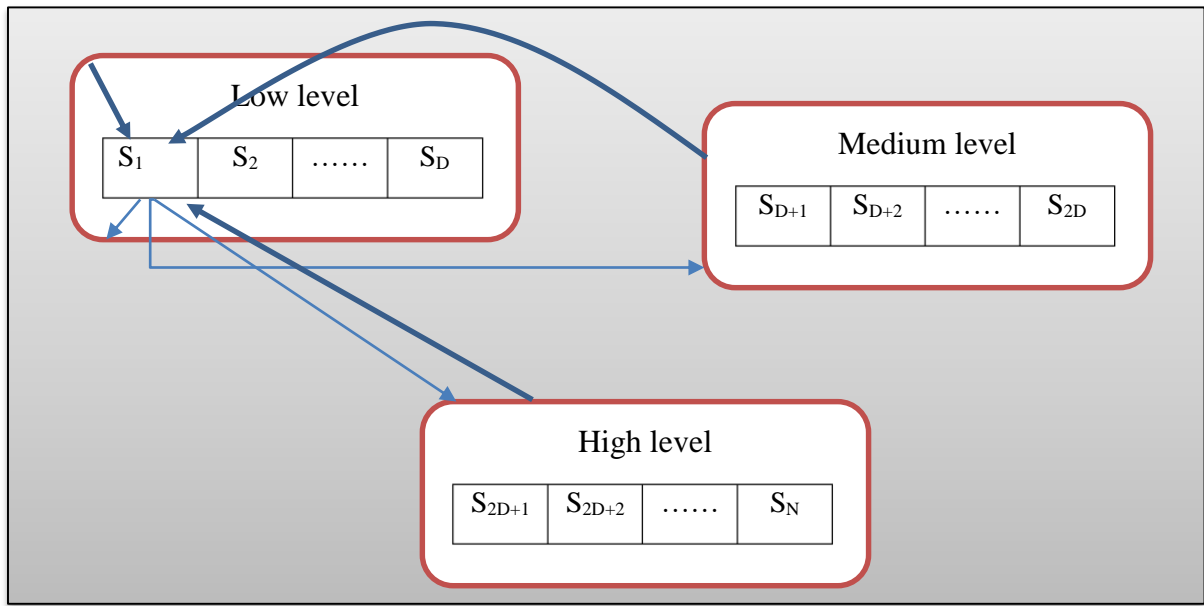


Figure 18: Students performance levels and reviewers distribution

The total number of students, N is divided into three levels, that is $D=N/3$ to make all levels have equal number of students. This is when the number of students is divisible by 3 otherwise the high level will have the remaining students. According to the requirements from the users as well as the literatures, it was found that three reviewers make a relatively effective reviews for a single student and also reviewing three fellow students makes the reviewer learn from the process and it is not tiresome. So, in this system, a student in the low level reviews other three students, one from each level and also s/he is reviewed by three other students, one from each level. In this Figure 18, Student S_1 reviews three students one from each level and in return S_1 is reviewed by three students, one from each level. The smaller arrows show the groups the student S_1 is reviewing while bigger arrows show the reviewers for student S_1 .

After assigning three reviewers per student's assignment, every student will review three fellow students' assignments and in turn receive three reviews from other students who are not necessarily the ones s/he reviewed.

5.5 Random subproducts algorithm

Random subproducts is an algorithm known for the efficient construction of random groups [50] whenever needed from large group of elements. This allows efficient randomized algorithm of many group constructions [51]. In permutation groups, constructing efficient groups depends on the first constructing a powerful generating set. An alternative approach towards approximately random group elements [52] which is sufficiently general to work over arbitrary black box group. Random subproducts can be constructed without having or knowing a strong generating set. They were introduced for use in orbit calculation by Babai, Luks and Seress [53]. Their use in developing randomized algorithm for generating group computations were first described in [54].

Definition of Random Subproducts algorithm: Given a generating set

$$S = \{g_1 \dots g_r\}$$

for a group G , a random subproduct on S is an element w of the form

$$w = \{g_1^{e_1}, g_2^{e_2}, \dots, g_r^{e_r}\}$$

Where $e_i = 0$ or 1 with probability $\frac{1}{2}$.

Proposition 1. Let S generate a group G and let H be an arbitrary proper subgroup of G . Then a random subproduct w on S is not a member of H , with probability at least $\frac{1}{2}$ [50].

Proof.

Let $S = \{g_1 \dots g_r\}$. There is a largest i for which $g_i \notin H$ (since $H \neq G$). So, w can be decomposed into the form $w = ug_i^{e_i}v$, with $g_{i+1}, \dots, g_r \in H$. Hence $v \in H$.

We consider two cases. First, assume $u \in H$. With probability $\frac{1}{2}$, $e_i = 1$, and so

$w = ug_iv \notin H$. In the second case, assume $u \notin H$. With probability $\frac{1}{2}$, $e_i = 0$, and so $w = uv \notin H$.

With modification to the algorithm, it is able to assign reviewers from different performance level depending on the criteria performances. This algorithm is more effective in the assigning of reviewers in this project most especially in the generation of the passing criteria.

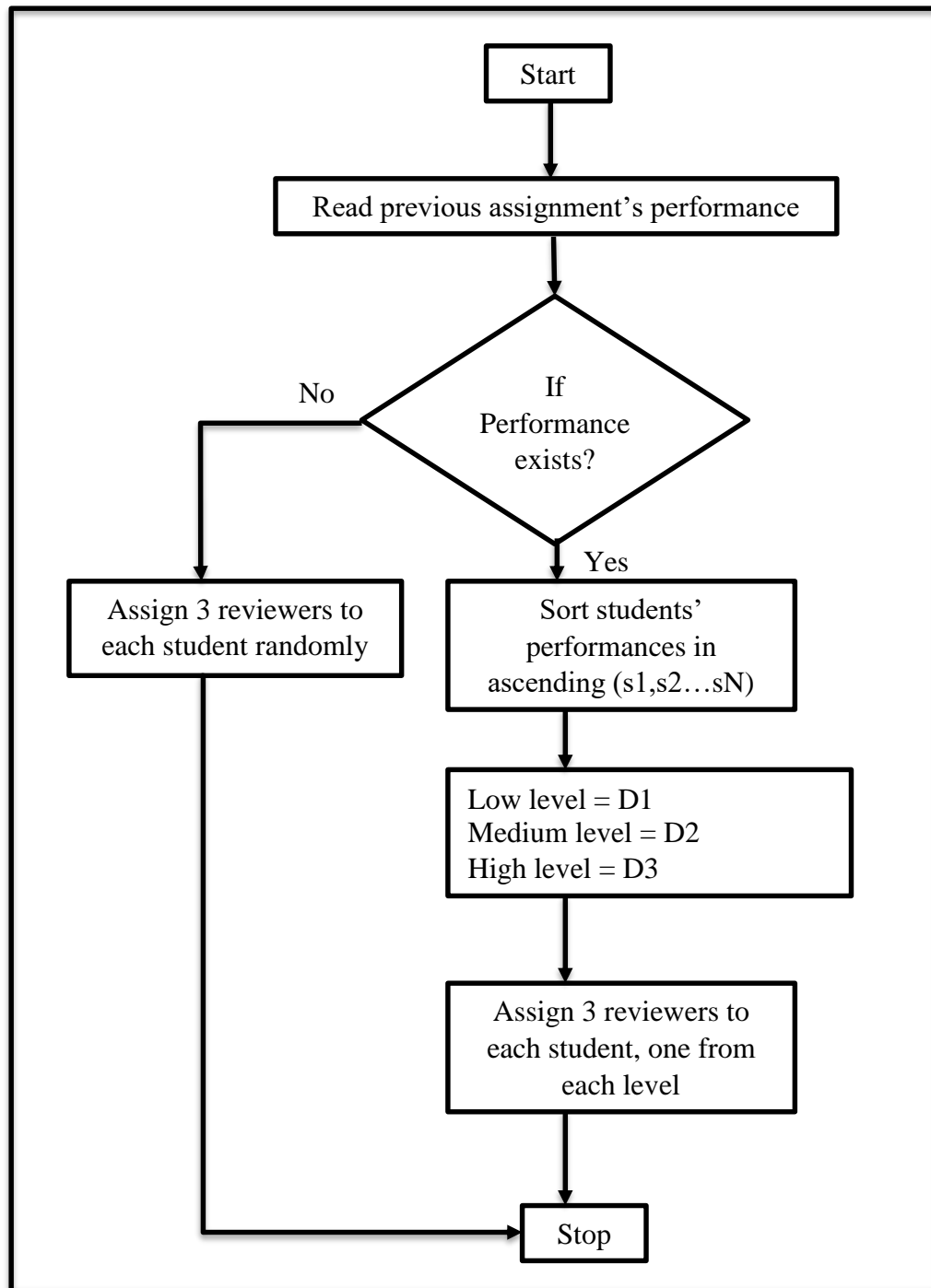
Peer matching flowchart

The Figure 19 shows the flowchart for the algorithm. This algorithm will be designed to allow all number types for the student's number, in case where the number of students is not divisible by three, the low and medium level will be assigned equal number of students and the high level will be assigned with the remaining number of students.

Students after being assigned to different groups, they will be given reviewers for one assignment which is the current assignment. This will be based on the previous assignment's performance if any, otherwise the reviewers will be assigned randomly three to each student as shown the Figure 19.

The legend in the Figure 19, explains the formulas in detail. $D = \text{total number of students} / 3$. For example, with 100 students sorted in ascending order, $D = 100 / 3 = 33$. The first 33 students belong to the low level which is named as D1, the medium level=D2 starts from the 34th up to 66th student. The remaining students from the 67th to the 100th belong to the high level which referred to as D3. Every student must receive at least three reviews and also review other but not limited to three assignments.

Algorithm process diagram



Legend

Low level = D1
 Medium level = D2
 High level = D3
 $D = S_n/3$
 $D1 = S_1 \dots S_D$
 $D2 = S_{D+1} \dots 2D$
 $D3 = S_{2D+1} \dots N$

Assuming $S_n=100$
 $D=S_n/3$ (Dividing total students into 3 groups e.g. $100/3=33.*$)
 $D1 = S_1 \dots S_{33}$ (e.g. Student 1...Student D)
 $D2 = S_{34} \dots S_{66}$
 $D3 = S_{67} \dots S_{99}$ or
 $D3 = S_{67} \dots S_{100}$

Figure 19: Peers matching process

5.6 Assessment criteria

Assessment criteria are the evaluation indicators that either students or teachers are using in the evaluation of the assignment tasks submitted by the students in the submission to the assignment task. They are assigned to as assignment depending on the requirements for that specific assignment. Only teachers can assign these assessment criteria to the assignment. Each criterion has weight which will be used in the classification of the students in different levels. The weights are determined by the teacher while setting the criterion, and will not be available or visible to student when reviewing the assignment.

In this project, assessment criteria were obtained from teachers, students and the literatures about learning and studying programming courses. Teachers provided their contribution in the forming of the assessment criteria framework through the interview conducted as mentioned in the previous section. Also students were given opportunity to express their inputs through a survey as mentioned in the previous section too. The literature about the learning and teaching of programming language courses included the IBM developer team [44], the bloom's taxonomy [2] and other related literatures as mentioned in the state of the art section. These criteria will be used to make sure that students know what the crucial parts of the programming language courses are. Table 4 shows the assessment criteria that can help in the assessment of programming assignments.

Table 4: Assessment criteria [44]

| No | Criterion | Description |
|----|-----------------------------|--|
| 1 | Using meaningful data types | Using real data types that are compiler recognized |
| 2 | Meaningful variable naming | Variables with meaning |
| 3 | Syntax | Syntax of the program |
| 4 | Condition statements | Conditional statements (e.g. If, switch, etc.) |
| 5 | Looping statements | Looping in the code |
| 6 | Using Classes | Program Classes defined by users |
| 7 | Using Methods/Functions | Program Methods/functions defined by user |
| 8 | Applying Comments | Comments in the program for communication purpose |

| | | |
|----|-------------------------|--|
| 9 | Correctness | Check if the final target is met by the code |
| 10 | Relevant Lines of code | Number of code should be relevant to program |
| 11 | Coding style | Style of the code (e.g. indentation) |
| 12 | Code readability | Code must be readable |
| 13 | Operations | Operations must be used correctly |
| 14 | Data input and output | Inputs and outputs for the code |
| 15 | Recursion | Recursive codes must be used appropriately |
| 16 | Library functions | Import library functions when necessary |
| 17 | Command line | Command line should be used if necessary |
| 18 | Arrays | Arrays must be used appropriately |
| 19 | String | String and characters must be used accordingly |
| 20 | Files | Files handling and usage |
| 21 | User defined data types | Users apply their defined data types |

These criteria were selected to help in the assessment of the introductory programming course being taught at the University of Agder at bachelor level but later can be applied to the rest of programming courses as they all share structure and sub topics.

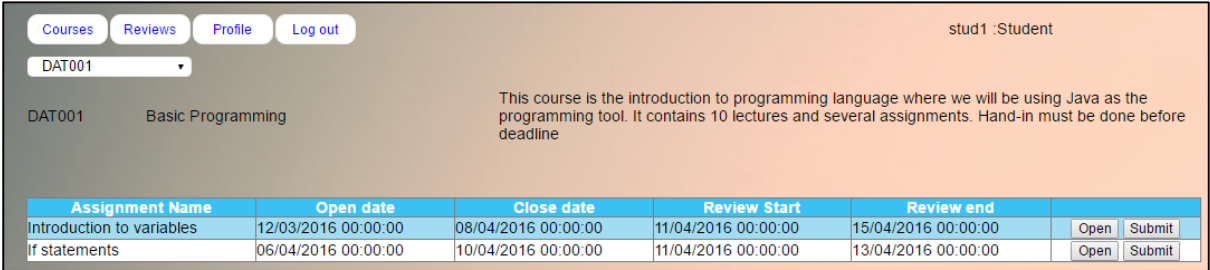
5.7 Interface design

The graphical user interface is the most useful in the creation and designing of application or system [55]. They must be designed in collaboration with the system users for the system being designed. The main users of the peer review system are the students who will be using the system to submit their solutions as well as review their peers and receive feedbacks from other students. The users played an important role in the development of the interfaces where they were approached in three iterations in different periods in order to get more efficient and understandable interfaces.

First iteration

Students were approached for the testing of the interfaces, for the first time, they found the interfaces to be confusing and not understandable. After we explained the interfaces and the expected use of the interfaces, they were happy to add more features that were not in the system interfaces. Starting from the login page, students recommended to use strong passwords so that the security of the system should be considered. We informed the students that the system we are developing will not be standalone system, it is a complement to the existing system with strong security policy. They agreed on the response we gave and continued to the next page.

The next page is a page showing the courses with the related assignments. To find a course, students were requested to enter the course name and it appears. They denied the search feature and recommended to use a dropdown list showing the courses available for that particular student. We agreed on the feedback and changed to the dropdown list. On the same page, students were able to find minor errors, we had used terms which they did not agree with. The terms were “Download” and “Upload correction” which mean to open the assignment and submit assignment solution. They recommended to change the terms to their meaning instead and we replaced with the terms to “Open” and “Submit”. The first iteration, as shown in Figure 20 was completed on the first page and we were able to apply the recommended changes.



The screenshot shows a web interface for a student. At the top, there are navigation buttons: "Courses", "Reviews", "Profile", and "Log out". The user is logged in as "stud1 :Student". Below the navigation, there is a dropdown menu showing "DAT001". The course details for "DAT001" are "Basic Programming". A description of the course is provided: "This course is the introduction to programming language where we will be using Java as the programming tool. It contains 10 lectures and several assignments. Hand-in must be done before deadline". Below the course details, there is a table with the following columns: "Assignment Name", "Open date", "Close date", "Review Start", "Review end", and two buttons: "Open" and "Submit".

| Assignment Name | Open date | Close date | Review Start | Review end | Open | Submit |
|---------------------------|---------------------|---------------------|---------------------|---------------------|------|--------|
| Introduction to variables | 12/03/2016 00:00:00 | 08/04/2016 00:00:00 | 11/04/2016 00:00:00 | 15/04/2016 00:00:00 | Open | Submit |
| If statements | 06/04/2016 00:00:00 | 10/04/2016 00:00:00 | 11/04/2016 00:00:00 | 13/04/2016 00:00:00 | Open | Submit |

Figure 20: First iteration of interface design

Second iteration

In the second iteration, the same students were approached but this time they were more than before in numbers. They were shown the corrected interfaces according to their first recommendations and they found them to have been done accordingly. On this page, a student can only choose a course s/he needs to view the attached assignments, view assignments, open assignment and submit solution to the assignment. They agreed on the page and we progressed to the Review page. As the same as the first page, the page contains a dropdown list as suggested by the students on the first page.

Students were able to find the course, all the assignments about the course were listed in the following list. They could easily select whatever assignment they need. After selecting the assignment, there appeared the list of three students that they were going to review but at the time, the review process was not active. They were shown the review page with the assessment criteria that will be used in the review process and agreed. They recommended us to add links directing to the review process and the one directing to the feedbacks received. Also they recommended to view code in the browser on the same page to simplify the overhead of opening many applications. According to their recommendation the interface was updated, as shown in Figure 21.

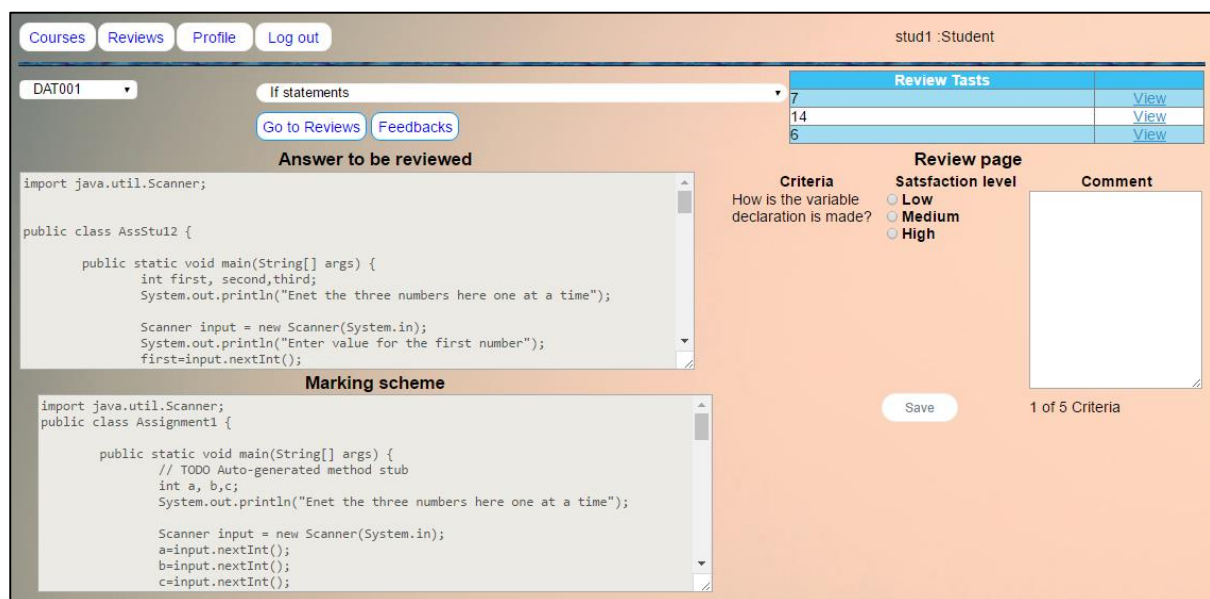


Figure 21: Second iteration of interface design

Third iteration

This was the most complicated task where we were doing the last part to be evaluated by the students on the interfaces to be used in the system. After developing the interfaces with all the recommendations from users, we developed the final interfaces, gave the option to open the files in the browser but without modification, added the links to the tasks to be reviewed and to the feedback received by the student. We had to show what we have done to students so that they confirm that what we made were based on their recommendations. They agreed on the features and requested to add the help information on the first page so that it will not be blank. We did the correction and came up to the final interfaces as shown in Figure 22 and Figure 23.

After designing the interfaces for students, also interfaces for teachers were designed with the minimum requirements. Teachers are able to add courses to the system, add assignment to the added course. Assignment contains the questions, solution to the question (reference

document), assessment criteria and important dates. The dates are the opening date for submission, closing date, review starting date and the closing date. Teachers are able to make review for any selected student and also view feedback for any selected student.

Courses **Reviews** **Profile** **Log out** stud1 :Student

Welcome to the Peer Review System.

You need to select a course you wish to work on first. Go to Courses menu and select the course from the dropdownlist. You will see the Assignments related with that course. As a student, you will be able to open the assignment and submit the solution to that assignment. To review others' work, go to Review menu, select the course from the dropdownlist and select an assignment. Click Go to Review you will get the list of students you are assigned to review, by clicking on the Make Review button, You will be given the solution for the selected student. Then evaluate the solution according to the criteria given at the right side of the page. When you are done with giving review to others, You may click on Feedbacks button and see what feedbacks were given to your solution too.

Thanks.

Figure 22: Third Iteration-information page

Courses **Reviews** **Profile** **Log out** stud1 :Student

DAT001 If statements

Go to Reviews **Feedbacks**

Answer to be reviewed

```
import java.util.Scanner;

public class AssSt5 {

    public static void main(String[] args) {
        int a, b, c;
        System.out.println("Enet the three numbers here one at a time");

        Scanner input = new Scanner(System.in);
        System.out.println("Enter value for the first number");
        a=input.nextInt();
    }
}
```

Marking scheme

```
import java.util.Scanner;
public class Assignment1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int a, b, c;
        System.out.println("Enet the three numbers here one at a time");

        Scanner input = new Scanner(System.in);
        a=input.nextInt();
        b=input.nextInt();
        c=input.nextInt();
    }
}
```

| Review Tasks | |
|--------------|--------------------|
| Student: 1 | Make Review |
| Student: 2 | Make Review |
| Student: 3 | Make Review |

Review page

| Criteria | Satsfaction level | Comment |
|--|---|---------|
| How is the variable declaration is made? | <input type="radio"/> Low <input type="radio"/> Medium <input type="radio"/> High | |

Save feedback 1 of 5 Criteria

Figure 23: Third Iteration-review process page

5.8 Personalised learning

The basic idea of Personalized Learning (PL) is the belief that each student is unique and learns in different ways. It has been suggested that personalized learning is originated from Howard Gardner's theory of multiple intelligences [56] [57].

Students may vary in their perceptions, motivations, and judgments about themselves, which can affect their approach to excel in various types of challenges. The growing demand of individual preferences and aptitudes, requires focus on personalized learning. It is where a student gains knowledge according to the variables including individual pupil interests, needs and abilities and the identification of the best learning style for each pupil [58].

In most of the educational systems, traditional teaching methods are used to provide same instructions to every student of the class, whereas PL aims to develop solutions which can for generate individualized lesson plans which are based on student's performance.

Many technology-based approaches to PL amount to nothing more than personalizing the reading of texts to students of different abilities—rather than personalizing a mix of activities that give students a richer and more meaningful educational experience, said Elliot Soloway, a professor of computer science at the University of Michigan who has studied and developed digital education tools.

Based on this, we introduced a system which offers a step towards personalized learning. We believe that the variety in individual's learning preferences demands optimized solutions to meet the students' requirements. Today, many web based educational systems are becoming popular. These educational systems are developed interactions to facilitate the students, sometimes with additional multimedia features.

The designed system will apply PL methods specifically to overcome the challenges faced in the learning of computer programming languages. The students will benefit when they receive their personalized feedback to the solutions they submitted. The mixed feedback from different performance levels and groups will make the feedback personalized for every student. Such learning methods can be highly motivating for the students.

The peer matching algorithm is further refined to fulfil the individual needs to learn and improve their programming skills. The delivery of the proper feedback is an important part of PL. This can only be possible with efficient and effective feedback for the students of different skill levels. The success of personalized learning is actually when we choose our own goals and we work harder to achieve them. Personalized learning was further used to develop a personalized peer matching algorithm. This algorithm was used to distribute reviewers to students based on individual performances in different assessment criteria.

Personalized peer matching algorithm

As peer matching is used to distribute reviewers among the students, it was found to be more helpful to distribute reviewers based on individual needs. Students have different capabilities in solving the given tasks and also the reviewers should be based on their needs. Personalized peer matching uses the total performance together with the individual criterion performances obtained by the student in the previous assignment.

From Table 5, students $S = \{s_i \dots s_N\}; i = 1, 2 \dots N$, students are classified in levels so that reviewers will be distributed from all levels, $S_l, l = \{Low, Medium, High\}$. Students receive and review three students from all levels. Every student is reviewed by three students all giving acceptance level and comment to the assessment criterion. All criteria have weights which are assigned by the teacher. Table 6 contains assessment criteria with their weight for explanation of the algorithm.

Table 5 shows students in different levels according their performances.

Table 5: Students in levels with their corresponding scores

| Student Number(S#) | Assessment criteria scores (P) | | | | | Total score (P) (%) | Level (S) |
|--------------------|--------------------------------|-----------------|-----------------|-----------------|-----------------|---------------------|-----------|
| | Criterion 1 (%) | Criterion 2 (%) | Criterion 3 (%) | Criterion 4 (%) | Criterion 5 (%) | | |
| S ₈ | 8 | 12 | 24 | 7 | 14 | 65 | Low |
| S ₃ | 8 | 14 | 21 | 7 | 16 | 66 | |
| S ₁₃ | 14 | 16 | 21 | 7 | 18 | 76 | |
| S ₁₁ | 14 | 18 | 24 | 7 | 16 | 79 | |
| S ₄ | 16 | 18 | 21 | 7 | 18 | 80 | Medium |
| S ₅ | 14 | 14 | 27 | 8 | 18 | 81 | |
| S ₉ | 14 | 16 | 24 | 7 | 20 | 81 | |
| S ₁₀ | 14 | 16 | 27 | 7 | 18 | 82 | |
| S ₁₂ | 14 | 14 | 30 | 7 | 20 | 85 | High |
| S ₁₄ | 20 | 20 | 27 | 8 | 20 | 95 | |
| S ₆ | 12 | 22 | 29 | 11 | 24 | 99 | |
| S ₇ | 12 | 22 | 32 | 9 | 23 | 100 | |

Table 5 shows the summary of the sorted performances obtained by students in the assessment criteria shown in Table 6.

Table 6: Assessment criteria with corresponding weights

| Criterion No | Description | Weight (%) |
|--------------|--|------------|
| 1 | How the variable declaration is made? | 20 |
| 2 | Are the inputs well stated | 20 |
| 3 | How is the “selection statement” used? | 30 |
| 4 | How are the comments made? | 10 |
| 5 | Is the overall program working? | 20 |

Criterion score in Table 5, Criterion# score,

$$C_i = 1/n \sum_{j=1}^n P_j, i = \{1,2,3,4,5\} \text{ and } j = \{1,2,3\}.$$

This means that there are 5 criteria each reviewed by 3 reviewers

The total score for all assessment criteria for one student,

$$P_r = \sum_{k=1}^n C_k, k = \{1,2,3,4,5\}, r = \{1,2, \dots, N\}.$$

Each student scores P_r points from all criteria as the overall performance in the assignment.

Peer matching algorithm compares student (as a reviewer)’s total performance with the author’s total performance and also the reviewer should not contain the same performance points on the same criterion as the author except if there is no other reviewer remaining. Thus, reviewer performance,

$$P_{rl} \neq P_{jl}, \text{ and } C_{ril} \neq C_{rjl} \text{ where } i, j = \{1,2, \dots, N\} \text{ and}$$

$$l = \{low, medium, high\}.$$

The overall personalized peer matched students can be found in Table 8.

Hence, personalized peer matching allows students be reviewed by any student who obtained a performance score different from the author’s performance by comparing the total performance and the individual assessment criteria. This helps student to get help in their weak points in the programming codes being reviewed and also reject all review where the author

and the reviewer have the same weaknesses. This is more important in the learning based on individual capability and fixing individual needs. The long code implementing the personalized peer matching is shown in the Appendix D.

6. Implementation and validation of peer review system

Peer review which is developed in this project work is an additional feature for existing tool, T-FLIP. The solution developed is based on the needs from the teachers and the students as mentioned earlier. This is made up of the interfaces that will be used for the review process, the peer matching algorithm and the assessment criteria. The peer matching algorithm and the assessment criteria are the most important parts of the project. Peer matching algorithm is used to make learning more personalised. This is accomplished by assigning reviewers based on the assessment criteria performances in combination with the overall performances in all the criteria as detailed in the personalised learning section. This solution (peer review), the implementation is composed of two parts: the teachers' component and students' component. These components are used by their respective owners (teacher and student respectively) as detailed in the implementation section below. All these parts were developed based on the results obtained from interview with teachers, survey from students and the literatures.

6.1 Design phase

The design phase is made of understanding and perspectives, where design and evaluation need to be undertaken. In the later section, scenarios of use will be developed and be used in the user need analysis. These will lead to the peer review requirements and also help in the validation of prototype.

In this phase we explain the design activities which will include both conceptual design and physical design. Conceptual design is about designing a system in the abstract whereas the physical design is concerned with making things concrete. Although, there is no clear concept to differentiate conceptual design or physical design but rather there are degrees of conceptuality [59].

Conceptual model

A typical conceptual model is a set of abstract and general statements about the concepts. These concepts are defined and their relationships in between them are expressed. One way to conceptualize the main features of the system is to use a rich picture. A rich picture captures

the main conceptual relationships between the main conceptual entities in a system [59]. The model helps to observe, understand and interpret the concept.

Figure 24 describes the working organization of the T-FLIP tool containing the peer grouping and peer review features. In the flow, it shows the users of the system who are the teacher and a student. They login or register for the system for the first time, a teacher is able to add a course to the system. After adding a course, the teacher is able to add an assignment related with that course. An assignment is added with the marking scheme and the assessment criteria that will be used during the peer review process.

After a student has logged in, s/he can register for an added course and also be able to download any assignment related with the course. S/he solves and upload the work to the system where is able to run the evaluation of the assignment. The submitted work will be compared to the marking scheme and the system gives the feedback to the student, the performance is saved. The feedback here is the pass or fail according to the T-FLIP tool.

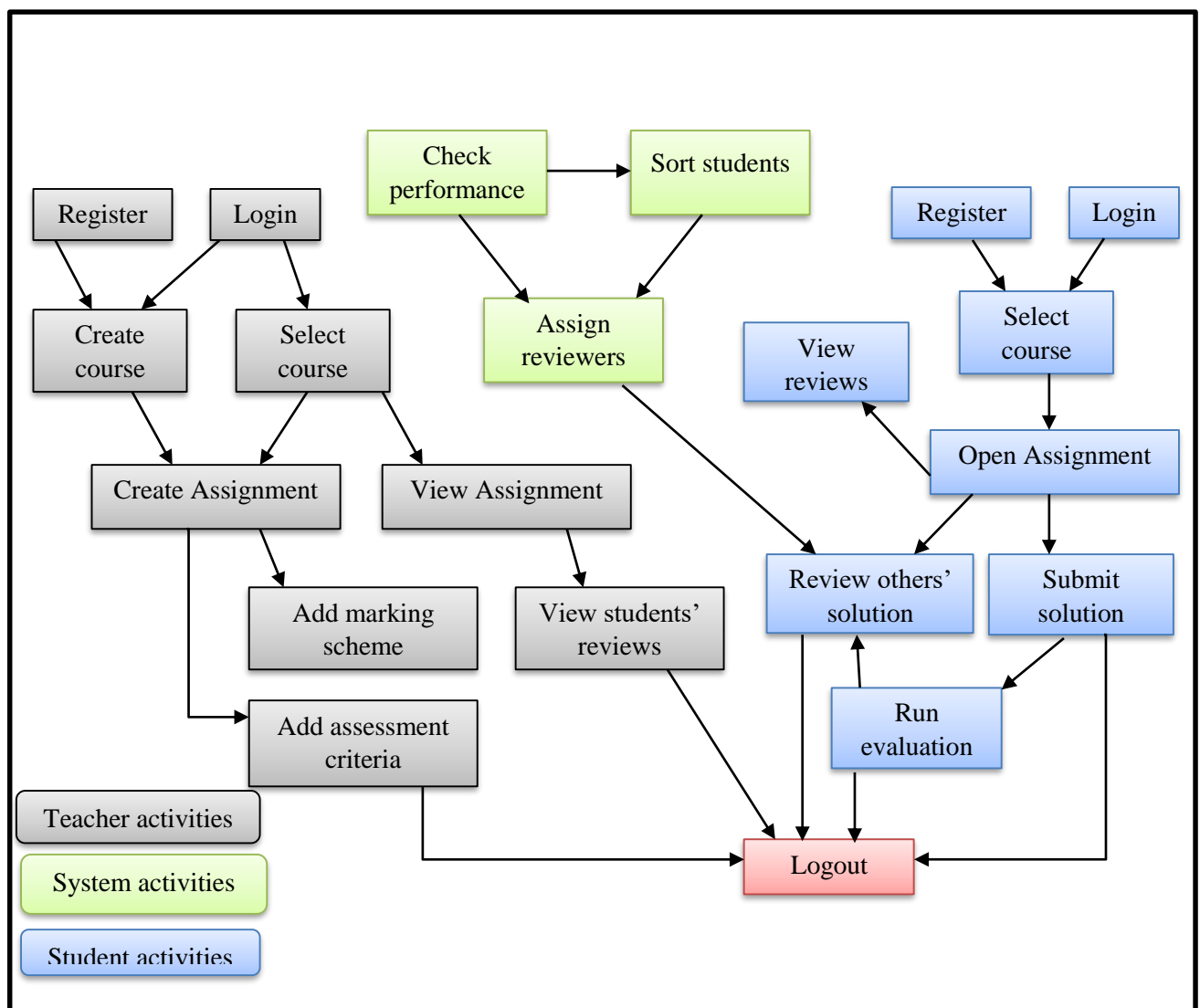


Figure 24: Conceptual model for peer review

Physical design of peer review system

The physical design explains how the system will work and details about the look and feel of the product. It is very important to understand the difference between physical and conceptual design. The conceptual design relates to the overall purpose of the whole interactive system between the people and the technologies. The physical design takes the abstract representation and translates it into solid designs. It needs to clarify the hardware and software requirements and also the tasks and activities the users will perform [59].

Peer review system architecture

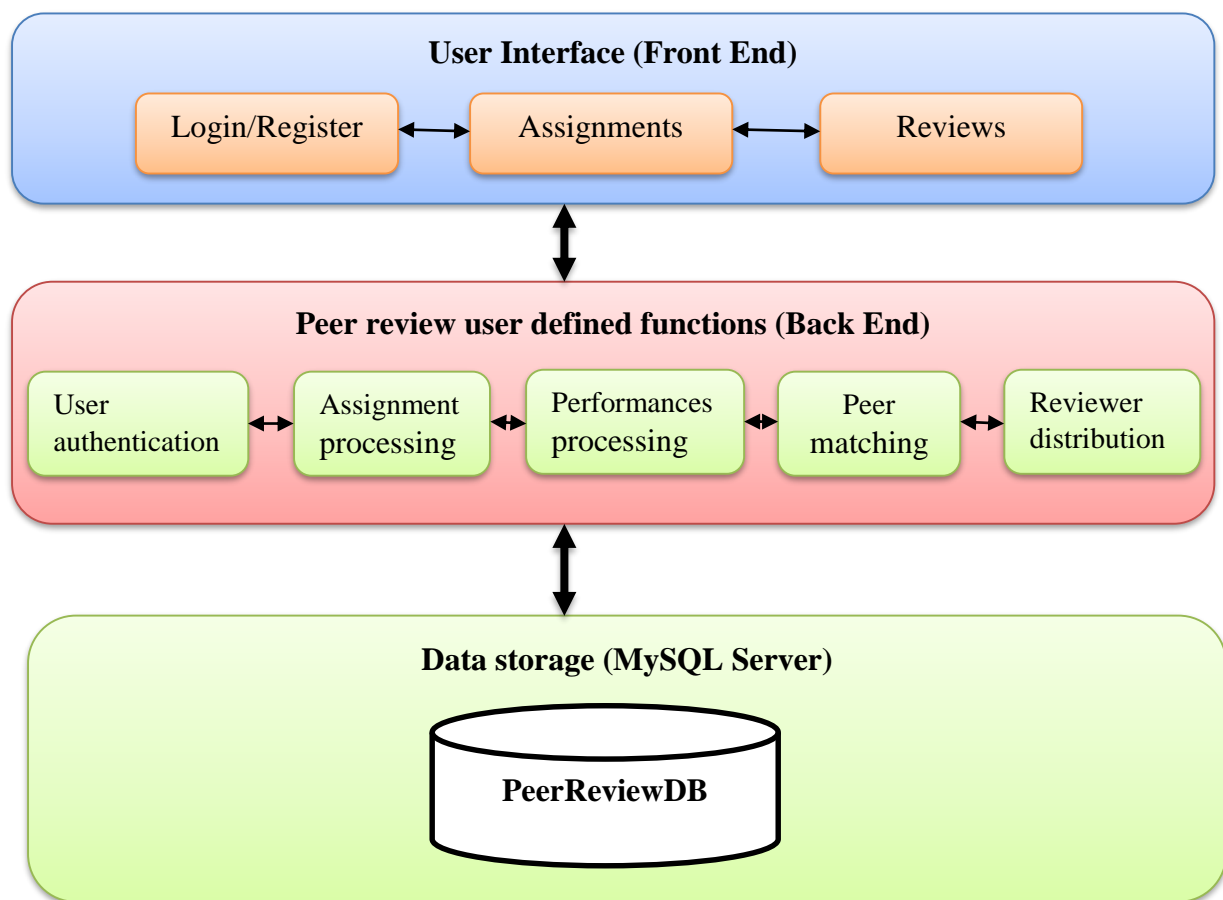


Figure 25: Peer review process architecture

Figure 25 presents system architecture for the system doing the peer review process as follows:

The first component is the MySQL server, is used to accommodate the database that stores all the data from the peer review system. A database "PeerReviewDB" is used as the storage for the data. The data being stored is the users using the system, the courses being assessed, the assignments tasks attached to the courses, solutions to the assignment tasks, reviews (feedback) to the solutions and the performances for the students that are obtained from the reviewed solutions.

The second component is the main components of backend, which is composed of the functional sub-components of the system. These sub-components are:

(1) *Peer matching* is the algorithm part that matches the peers as reviewers. As described in Figure 25, the algorithm assigns reviewers to each student from three different levels where each student gets at least three reviewers and in turn s/he gives back three reviews to the fellow students.

(2) *Reviewers distribution* is the component responsible for the distribution of the reviewers assigned to the author of the assignment answer by the peer matching component.

(3) *Assignments processing* is the component responsible for the assignment handling. This is composed of the submission of the assignment by the teacher or the instructor and the submission of the student's solution to the assignment. Any process concerning assignments is combined in this component until the saving of the assignment in the database. Also the retrieval of assignments by users is included in the assignment processing.

(4) *Performances processing* handles the retrieval and saving of students' performances from/to the database and can be used by any component that needs them. One example is the peer matching component that needs the performances in order to sort the students into three performance levels.

(5) *User authentication* is the component that is responsible for making sure that all users are checked before accessing the system and granted the access when they are found to be registered, otherwise rejected. It compares the credentials entered by the user with the existing and saved credentials in the PeerReviewDB.

The third main component of the architecture is the user interface (front end) which is composed of different interfaces.

(1) *Login/ Register* is one of the interfaces that can be found in this component and it allows the user to access the system, so that the system can differentiate between its users. It also allows the new users to register to the system and become system users. This component connects to the user authentication through the backend and verifies the credentials.

(2) *Assignments* is an interface that makes one of the core parts of the system where it displays all the assignments that are given within a selected course. Here the students can access the assignments, download them for solving and finally submit the solution to the assignment interface. This interface connects to the assignments processing component in the peer review

internal components through the backend, where the assignment information is finally saved to PeerReviewDB.

(3) *Reviews* is the interface that shows all information regarding the reviews, either the reviewing process or the reviews received from other peers. The interface connects to the reviewers' distribution component in the backend where the included information is saved to the database.

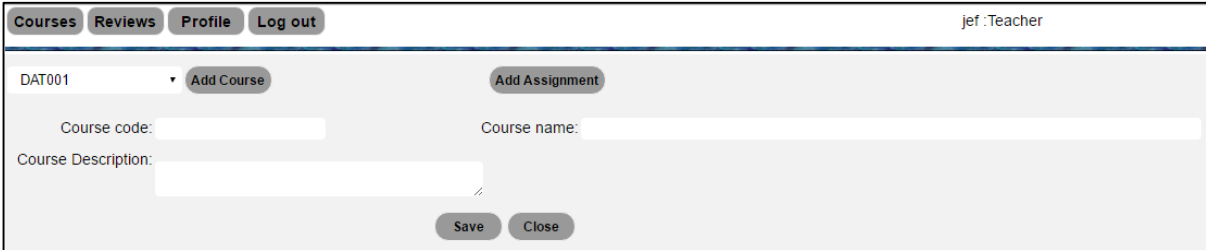
6.2 Implementation

The system functionality starts by the user login. Users are registered and used their usernames and passwords to login to the system. Security is one of the ignored parts in the development of this component, this is because, and it has been developed and considered in the development of T-FLIP. Despite of ignoring the security part, users must be registered for them to use the system. There are mainly two types of users but they will be facilitated by the administrators of the system. Only students can create their user accounts and after creation, they are ready to use. Teachers need to be given permissions from administrator to use the account as a teacher otherwise s/he will be registered as a student.

Teacher's activities in the peer review system component

Starting from the teacher's profile, a registered teacher is able to add a course to the system add assignment with assessment criteria as shown in the Figure 26.

In the Figure 26, the teacher gives details of the course; course code (e.g. DAT001), the course name (e.g. Basic Programming) and the course description for clarification to any one that come across the course. Teacher after filling the form, s/he saves the course but also with an option for closing the form if it was mistakenly opened.



| Courses | | Reviews | Profile | Log out | jef :Teacher |
|---------------------|---|--------------|----------------|---------|--------------|
| DAT001 | ▼ | Add Course | Add Assignment | | |
| Course code: | | Course name: | | | |
| Course Description: | | | | | |
| Save | | Close | | | |

Figure 26: Add new course to the system

After saving the course, it will appear in the dropdown list. When the course is selected, it is displayed with the related assignments if any, otherwise no assignments will be displayed. As shown in the Figure 40.

When teacher have added a course, there can be an option for adding an assignment. In system testing, students were given a task to write a program that allows a user to enter an integer value, and the program should display the prime number less than the entered number. An interface for adding assignment is shown below.

The screenshot shows a web interface for a teacher to add an assignment to a course. At the top, there are navigation tabs: Courses, Reviews, Profile, and Log out. The user is logged in as 'jef :Teacher'. Below the navigation, there is a dropdown menu for the course 'DAT001' and buttons for 'Add Course' and 'Add Assignment'. The course details for 'DAT001 Basic Programming' are displayed, including a description: 'This course is the introduction to programming language where we will be using Java as the programming tool. It contains 10 lectures and several assignments. Hand-in must be done before deadline'. The 'New Assignment' form includes the following fields: Name (text input), Description (text area), Open date and time (text input with a 'Date..' button), Close date and time (text input with a 'Date..' button), Review start date and time (text input with a 'Date..' button), Review end date and time (text input with a 'Date..' button), Upload Question file (Choose File button, No file chosen), and Upload Marking scheme (Choose File button, No file chosen). At the bottom of the form are 'Save' and 'Close' buttons.

Figure 27: Add an assignment to DAT001 course

From Figure 27, the teacher adds the following:

The name for the assignment, this is the name that differentiates the assignment from others. This name must be available for both teachers and students while accessing the assignment.

Description of what must be in the assignment if not specified in the assignment question file.

The open date and time, the time for the assignment to be ready for students to submit their solutions. Close date and time is the deadline for submitting the assignment solution.

Upload Question file is a field that accepts an assignment question file. The file must be of any type and will be opened and downloaded by the student while it comes to task solving time. The assignment question file is shown in the Figure 44.

Upload Marking scheme is a field that will be used to upload the reference document which will be used in the review process. This document will not appear anywhere before review time reaches. During the review time, the document will be viewed by both students and teachers. The reference code file is shown in Figure 45. After adding assignments to a course, they will be displayed as shown in Figure 41.

After saving the assignment to the system, the teacher must also save the assessment criteria that will be used in the review process. A criterion must contain a description and a weight. The description will be available in the review process while the weight will be used in the

classification of the students while forming reviewers for the next assignment. The process for adding criteria is shown in the Figure 42.

After giving the assessment criteria to the assignment, the assignment is ready to be reviewed when the review time reaches. The teacher is able to view the students' reviews from all the reviewers (feedbacks) and also can make a review for a student in case it is needed. Figure 28 shows the interface for the teacher after the review is made.

| Students | | |
|----------|-------------|---------------|
| stud1 | Make Review | View Feedback |
| stud2 | Make Review | View Feedback |
| stud3 | Make Review | View Feedback |
| stud4 | Make Review | View Feedback |
| stud5 | Make Review | View Feedback |
| stud6 | Make Review | View Feedback |
| stud7 | Make Review | View Feedback |
| stud8 | Make Review | View Feedback |
| stud9 | Make Review | View Feedback |
| stud10 | Make Review | View Feedback |
| stud11 | Make Review | View Feedback |
| stud12 | Make Review | View Feedback |

Figure 28: Teacher view on students review and feedback

In Figure 28, the teacher possesses two options for the student; “Make Review” and “View Feedback”. Make Review is used to make a review for the specified student in case it is needed. Making review by the teacher follows the same process as the student does. View feedback is used to view the feedbacks for a specific student. All feedbacks will be listed as shown in the Figure 29.

While viewing the feedbacks for a specific student, there will appear the solution document for the specific student and the reference document can be viewed to check the correctness and verification for the feedback obtained. After viewing these feedbacks, the teacher can make a new review for that student. These are the actions which can be performed by the teacher in the current peer review system component.

| Solution to the assignment | | Criteria | Comments |
|--|--|--|------------------------------------|
| <pre>public class Stud1Prime { public static void main(String[] args) { int i =0; int num =0; String primeNumbers = ""; for (i = 1; i <= 100; i++) { int counter=0; for(num =i; num>=1; num--) { if(i%num==0) { counter++; } } } } }</pre> | | How are the data types and keywords are used in the program? | data types are well used |
| | How are the data types and keywords are used in the program? | well used but needs details. String is used instead of integer | |
| | How are the data types and keywords are used in the program? | string is used which is not good | |
| | How do the inputs and outputs used ? | no input are used | |
| | How do the inputs and outputs used ? | inputs not used, outputs well used | |
| | How do the inputs and outputs used ? | use inputs instead of initialization | |
| | | Is there any method used? how is it used | no method used |
| | | Is there any method used? how is it used | No method |
| | | Is there any method used? how is it used | Use method as described by teacher |
| | How is the communication used throughout the codes | no communication is shown in the code | |
| How is the communication used throughout the codes | communication is not enough | | |
| How is the communication used throughout the codes | use comments to explain what you are doing | | |
| Is the code readable and correct? | code readable and correct | | |
| Is the code readable and correct? | only displays from 1 to 100 instead of n | | |
| | Is the code readable and correct? | readable | |
| | | | |
| Reference document | | | |
| <pre>import java.util.Scanner; public class PrimeNumbers { public static void main(String[] args) { System.out.println("Enter the number till which prime number to be printed: "); // Message to tell the user to enter the number int limit = new Scanner(System.in).nextInt(); // input way for the number //printing prime numbers till the limit (1 to limit) System.out.println("Printing prime number from 1 to " + limit); for(int number = 2; number<=limit; number++){ //bounding all the numbers till limit //print prime numbers only } }</pre> | | Is there any method used? how is it used | Use method as described by teacher |
| | How is the communication used throughout the codes | no communication is shown in the code | |
| | How is the communication used throughout the codes | communication is not enough | |
| | How is the communication used throughout the codes | use comments to explain what you are doing | |
| | Is the code readable and correct? | code readable and correct | |
| | Is the code readable and correct? | only displays from 1 to 100 instead of n | |
| | | Is the code readable and correct? | readable |
| | | | |

Figure 29: Feedback given to a student by three reviewers

Student's activities in the peer review system component

As peer review system is mainly developed to improve students' learning, it is expected to help them to gain more programming techniques from their peers. Students have the main activities of the system. They read the assignment task provided by the teacher, they solve the task using their local compilers and submit the solution to the system. The submission is done within the time interval as specified by the teacher in the saving of the assignment. Students submit their solution files to the system through the submission interface as shown in the Figure 43.

After all students have submitted their solutions and the time for closing the submission reaches, the system assigns reviewers to every student. Table 8 shows the students with their assigned reviewers from the performance levels in Table 7.

Table 7: Students in their respective performance levels

| SN \ Levels | Low | Medium | High |
|-------------|-----|--------|------|
| 1 | 3 | 4 | 6 |
| 2 | 8 | 5 | 7 |
| 3 | 11 | 9 | 12 |
| 4 | 13 | 10 | 14 |

Table 8: Students assigned with reviewers

| Student ID | Reviewer 1 | Reviewer 2 | Reviewer 3 |
|-----------------|-----------------|-----------------|-----------------|
| S ₃ | S ₈ | S ₁₀ | S ₁₂ |
| S ₄ | S ₁₁ | S ₁₀ | S ₁₂ |
| S ₅ | S ₃ | S ₉ | S ₁₂ |
| S ₆ | S ₁₁ | S ₅ | S ₁₄ |
| S ₇ | S ₁₃ | S ₁₀ | S ₆ |
| S ₈ | S ₁₁ | S ₉ | S ₁₄ |
| S ₉ | S ₈ | S ₅ | S ₆ |
| S ₁₀ | S ₁₃ | S ₄ | S ₇ |
| S ₁₁ | S ₃ | S ₅ | S ₆ |
| S ₁₂ | S ₁₃ | S ₄ | S ₇ |
| S ₁₃ | S ₃ | S ₄ | S ₁₄ |
| S ₁₄ | S ₈ | S ₉ | S ₇ |

In Table 7, students are classified in performance levels according to the total performances in all assessment criteria and the individual criteria for the previous assignment. Table 9 containing the students' performances in the previous assignment. It contains the points each student scored on each assessment criterion, the total score on the criterion from three reviewers and the average points scored. It contains also the total average score for all the criteria. Students' numbers (StudentId) were automatically assigned to students in the registration. The table shows the students that are sorted and ready to be categorized in three different performance levels. The peer matching algorithm sorts the students according to all scores from individual assessment criteria and the total score as shown in Table 9.

Table 9 shows the students' performance levels with their corresponding scores obtained from the previous assignment. StudentId is the number assigned to student for identification, TotCrit1 is the total score on criterion 1 from all three reviewers on a specific student, AvgCrit1 is the average score on criterion1 by that student, Total is the combination of all averaged scores for all criteria and finally, the Level is the performance level in which a student is classified. Students from Table 9 are assigned with reviewers from different levels as shown in Table 8 in a way that no student can review his work neither a student can be reviewed by all students that have the same score in assignment.

Table 9: Students in levels with their corresponding scores

| StudentId | Assessment criteria scores | | | | | | | | | | Total | Level |
|-----------------|----------------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-------|--------|
| | TotCrit1 | AvgCrit1 | TotCrit2 | AvgCrit2 | TotCrit3 | AvgCrit3 | TotCrit4 | AvgCrit4 | TotCrit5 | AvgCrit5 | | |
| S ₈ | 24 | 8 | 36 | 12 | 72 | 24 | 21 | 7 | 42 | 14 | 65 | Low |
| S ₃ | 24 | 8 | 42 | 14 | 63 | 21 | 21 | 7 | 48 | 16 | 66 | |
| S ₁₃ | 42 | 14 | 48 | 16 | 63 | 21 | 21 | 7 | 54 | 18 | 76 | |
| S ₁₁ | 42 | 14 | 54 | 18 | 72 | 24 | 21 | 7 | 48 | 16 | 79 | |
| S ₄ | 48 | 16 | 54 | 18 | 63 | 21 | 21 | 7 | 54 | 18 | 80 | Medium |
| S ₅ | 42 | 14 | 42 | 14 | 81 | 27 | 24 | 8 | 54 | 18 | 81 | |
| S ₉ | 42 | 14 | 48 | 16 | 72 | 24 | 21 | 7 | 60 | 20 | 81 | |
| S ₁₀ | 42 | 14 | 48 | 16 | 81 | 27 | 21 | 7 | 54 | 18 | 82 | |
| S ₁₂ | 42 | 14 | 42 | 14 | 90 | 30 | 21 | 7 | 60 | 20 | 85 | High |
| S ₁₄ | 60 | 20 | 60 | 20 | 81 | 27 | 24 | 8 | 60 | 20 | 95 | |
| S ₆ | 38 | 12 | 68 | 22 | 93 | 31 | 35 | 11 | 74 | 24 | 102 | |
| S ₇ | 38 | 12 | 68 | 22 | 102 | 34 | 28 | 9 | 74 | 24 | 103 | |

From the above information on the reviewer's assignment, students were able to find the authors they had to review. Every student reviewed three fellow students and was also reviewed by at least three students. The review process and the interface are shown in Figure 30.

Students are given the review interface shown in Figure 30, s/he selects a course from the list of course. The corresponding assignments are listed from where the student selects an assignment s/he is reviewing. On the selection of assignment, the student is given options from which to select either make review (Go to Reviews button) or view feedback (Feedback button). When a student enters the review process, there will be the list of students s/he must review or give feedback to their solutions, Figure 30, the section titled as "Review tasks" shows three students to be reviewed by the student. By selecting Make Review button after a student, the solution provided by that student appears in "answer to be reviewed section" with the reference code file in "Marking Scheme" section. In the "Review page" section appears the assessment criteria corresponding to that assignment.

[Courses](#)
[Reviews](#)
[Profile](#)
[Log out](#)

stud1 :Student

DAT001

Number Method: Prime number

[Go to Reviews](#)
[Feedbacks](#)

Answer to be reviewed

```

import java.util.Scanner;
public class Stud11Prime {

    public static void main(String[] args) {
        System.out.println("Enter the number till which prime number to be printed: ");
        int limit = new Scanner(System.in).nextInt();

        //printing primer numbers till the limit ( 1 to 100)
        System.out.println("Printing prime number from 1 to " + limit);
        for(int number = 2; number<=limit; number++){
            //print prime numbers only
            if(isPrime(number)){

```

Marking scheme

```

import java.util.Scanner;
public class PrimeNumbers {

    public static void main(String[] args) {

        System.out.println("Enter the number till which prime number to be printed:
"); // Message to tell the user to enter the number
        int limit = new Scanner(System.in).nextInt(); // input way for the number

        //printing prime numbers till the limit ( 1 to limit)
        System.out.println("Printing prime number from 1 to " + limit);

```

Review Tasks

| | |
|------------|-----------------------------|
| Student: 1 | Make Review |
| Student: 2 | Make Review |
| Student: 3 | Make Review |

Review page

| Criteria | Satsfaction level | Comment |
|--|---|---------|
| How are the data types and keywords are used in the program? | <input type="radio"/> Low <input type="radio"/> Medium <input type="radio"/> High | |

[Save feedback](#)
1 of 5 Criteria

Figure 30: Review process

The student reviews the solution according to the solution provided by the code author and the reference code file provided by the teacher. Review is made according to the assessment criteria by giving the acceptance level to the criteria and giving a comment about how the criterion is met by the author. After giving the comment in the comment-box, the student saves the feedback and get the next criterion. The student is notified with the review progress by getting the criteria number out of the total criteria. At the completion of reviewing one student, the student gets a message saying that it is complete, the student can start the review for the next student's solution.

When review process is done and the review period reaches, students will be able to view all feedbacks for their work. Here, students will be directed to the feedback page after pressing the Feedbacks button. The student is given option to view one review given from one reviewer at once as shown in the Figure 31. The student is provided with the list of reviewers where the student checks the feedback from that specific student without knowing who the reviewer was.

The screenshot displays a web application interface for a student named 'stud1 :Student'. At the top, there are navigation tabs: 'Courses', 'Reviews', 'Profile', and 'Log out'. Below these, the course 'DAT001' and the assignment 'Number Method: Prime number' are selected. There are buttons for 'Go to Reviews' and 'Feedbacks'.

The main section is titled 'Solution to the assignment' and contains a code editor with the following Java code:

```
public class Stud1Prime {
    public static void main(String[] args) {
        int i =0;
        int num =0;

        String primeNumbers = "";

        for (i = 1; i <= 100; i++)
        {
            int counter=0;

```

Below the solution, there is a 'Hide Reference document' button and a 'Reference document' section containing another Java code snippet:

```
import java.util.Scanner;

public class PrimeNumbers {

    public static void main(String[] args) {

        System.out.println("Enter the number till which prime number to be printed:
"); // Message to tell the user to enter the number
        int limit = new Scanner(System.in).nextInt(); // input way for the number

        //printing prime numbers till the limit ( 1 to limit)
        System.out.println("Printing prime number from 1 to " + limit);

```

On the right side, there is a 'Feedback Number' table with three rows for Reviewer 1, 2, and 3, each with an 'Open feedback' button. Below this is a table with 'Criteria' and 'Comments' columns:

| Criteria | Comments |
|--|--------------------------------------|
| How are the data types and keywords are used in the program? | data types are well used |
| How do the inputs and outputs used? | no input are used |
| Is there any method used? how is it used? | no method used |
| How is the communication used throughout the codes | no communication is shown i the code |
| Is the code readable and correct? | code readable and correct |

Figure 31: Feedback from reviewer 1

The student was given his/her solution, the option to view the reference code file and the feedback received from the reviewer. Students were expected to read the feedback given by reviewers and also compare with their solutions and the reference code file. In case the student is not satisfied with the feedback provided, s/he can contact a teacher and her/his solution can be reviewed by the teacher. Teacher review follows the same review process as the student does.

The development of the peer review system component aimed at making students learn from their fellow students' work while reviewing, give their perspectives about the solution being reviewed and also learn from the feedbacks they receive from other students [60], [61]. After completing the review process and the students received their feedback, they were satisfied and convinced with the feedback they got and found the system to be more helpful for learning programming course. The students were given a post-survey to express their perspectives about the system. They also recommended to use the system to other courses so that they can be assisted in the overall learning.

6.3 Validation

Validation is the process of testing and evaluating whether the developed software or system fulfil the requirements (IEEE Standard 1012-2004). Validation of the system can include the users of different types, here we can mention the end users, experts and the developers. This is a process of determining the correctness of the product based on the testing that can be done

using different techniques. In the validation of peer review system, we used usability testing approach to validate the system for its fulfilment about the requirements provided in the requirement specification section.

Usability has been defined in different ways based on the requirement. Semantically, it is a user friendliness of the system while in use. This is done by the end users testing whether the system will be easy to use. On features, usability is defined as the presence or the absence of some features that were expected to be in the interface. Operational usability is used to refer to the performance and effective levels as described by the users while doing a certain task using the system [62]. All these types of usability were considered in the developing of peer review system and a user based evaluation method was used in developing this system.

Evaluation methods

These are the methods that are used to validate the system for its usability. There are three methods which are used to evaluate and validate software or system, these are:

User-based methods: This is where a sample of intended users are used to test and evaluate the system by performing predefined tasks. This is considered to be reliable and its validity is considered. Users perform the tasks either in an environment similar to the real life of the system being tested in order to test whether the system supports the intended requirements. Users are intended to evaluate the system ease of use, features and operational approaches. Users provide their satisfactory recommendation after the completion of the testing which must be fixed by the development team [62].

Expert-based methods: This refers to the form of usability testing where an expert in the field is used to examine and estimate the likely usability of a given system. In this method, no end users are involved, the final evaluation is based on the judgement and interpretation of the evaluator. This method is preferred most as it provides quicker and cheaper results compared to user based [62]. In human computer interaction, there are two types of expert based evaluation; Heuristic evaluation and cognitive walkthrough. According to Nielsen [63], heuristic evaluation provides a set of evaluation guidelines the evaluators use to judge the system interfaces while performing a given task and reports violations to the guidelines. Cognitive walkthrough [64] evaluation is performed by an evaluator (expert) by examining the sequence of correct task performance and estimates the interfaces, likely success or failure of user performing the sequence [62].

Model based methods: These are the least used usability evaluation approaches but which are also proposed to be accurate for the prediction of time to complete the task by the users [62].

Validation of peer review

Peer review as a system component to help students in the peer review process, was validated through a user testing using User-Based Methods. Students were given a task to write a program that displays all prime number less than an integer number of their choice as shown in the Figure 44. Students were expected to make a code file using any development environment, upload the file to the peer review system through user accounts they were given at the starting of the evaluation. Creating their respective accounts were not considered crucial because peer review will be integrated with an existing system that already have registered students.

The most crucial parts to be evaluated were the reviewer distribution, review process and feedback receiving. These were selected based on the system's most important requirements. The reviewer distribution was checked to find if users were given to review their own assignment or any other peer matching violation was encountered. Students reviewed their fellow and gave them feedbacks as shown in Figure 30 describing the review process. After review process, students were requested to read the feedback they received from their reviewers and give recommendations. At the completion of the task, they were given a post-survey questionnaire, shown in Appendix C to express their experience to the peer review system during the process.

Students were asked if the peer review process was found helpful in order to learn programming courses based on their peers' solutions and feedbacks. 5 (41.7) students found it to be more helpful while 7 (58.3) out of 12 found helpful in the learning benefits. Students were asked this question to get their perspectives about the system and if they found to be helping.

They were given a second question about how easy was the review process while giving feedbacks to their fellow students. 7 students found it easy, 1 student found it not easy and 4 students found it too easy to give review to the fellow students using the peer review system as described in the Figure 32.

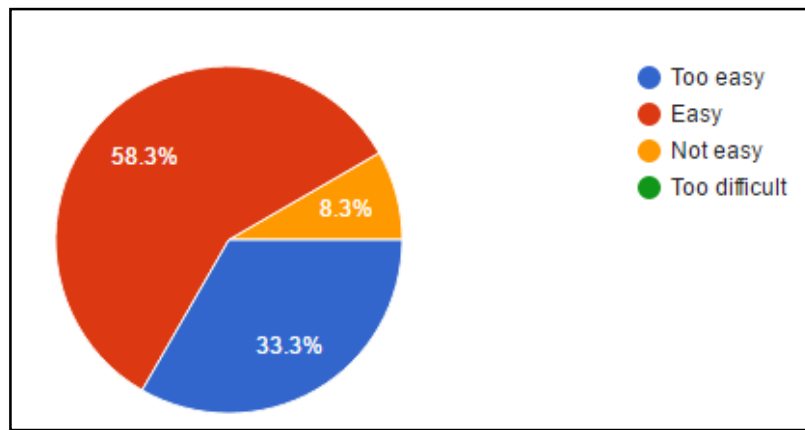


Figure 32: Post-survey on the difficulty of peer review process

They were asked this question to prove the difficulty of the process so that it can be improved but it was found to be generally easy for them to complete the review process. 1 student who found it not easy was asked about the difficulty and responded that it was browser that crashed and restarted itself while performing the task.

Thirdly, they were asked whether they learnt from reviewing others. 4 students agreed that they learnt from reviewing others while 8 strongly agreed to have learnt from reviewing their fellows as shown in Figure 33.

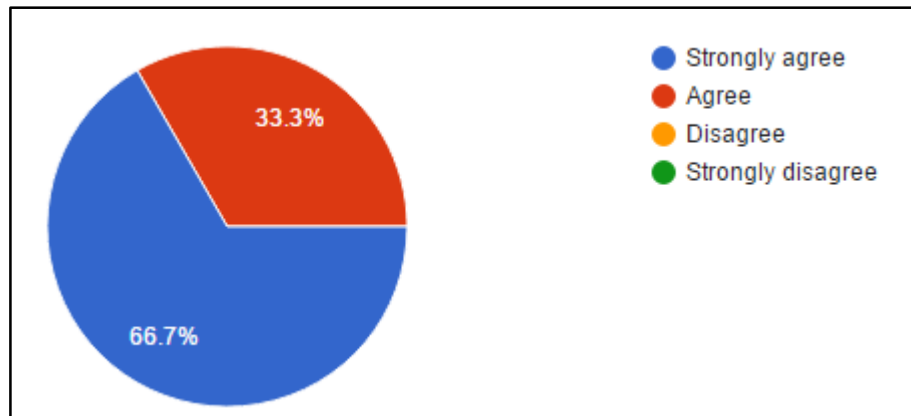


Figure 33: Post-survey on learning from reviewing process

Students agreed on average that the peer review system helped them to learn from the review process as they were looking at the codes which were done by other students.

The next question was about the distribution of the reviewers, 11 students responded to this question and 1 did not respond to this question. 7 students strongly agreed that the distribution were done based on the previous assignment's criteria performances, 4 agreed that it was also distributed based on personal performances as shown in Figure 34.

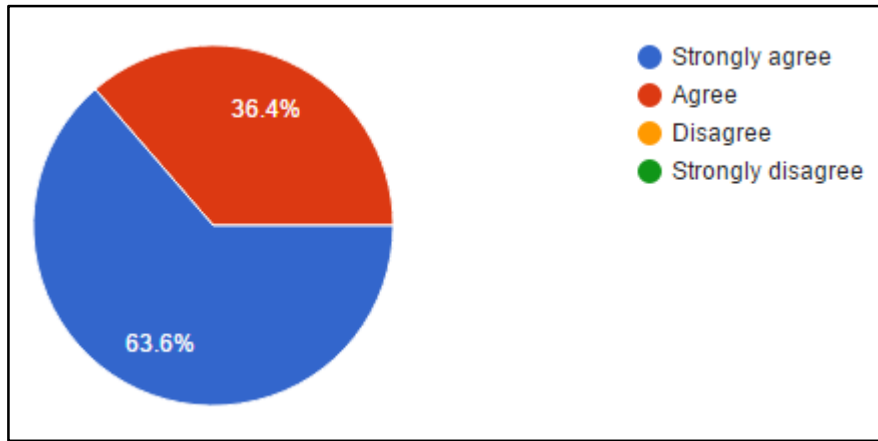


Figure 34: Post-survey on reviewer distribution

Distribution of reviewers were based on personal performances obtained in the assessment criteria for the previous assignment. The previous assignment was before the evaluated assignment and it was not evaluated for system validation but for having the predecessor assignment that can be based on to obtain personalised peer matching. The reviewers for the first assignment were distributed randomly from the students that submitted the solution to the assignment task. This was done in accordance with the peer review system requirements and peer matching functionality where the first assignment on every course is reviewed randomly by three reviewers per assignment solution.

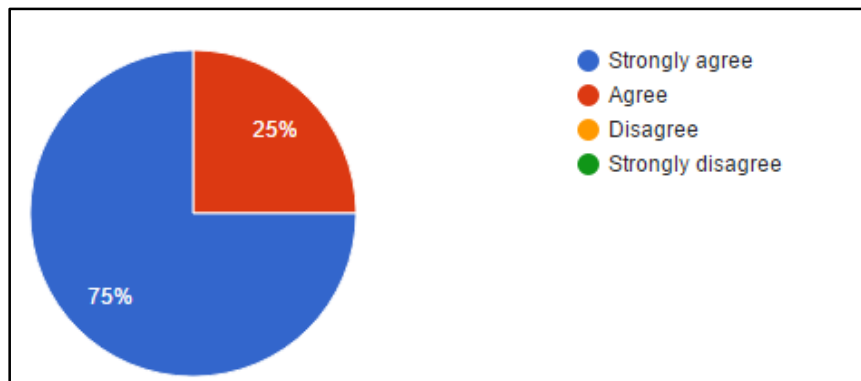


Figure 35: Post-survey on learning from feedback

Students were also asked the usefulness of the feedback they received from the reviewers. The main purpose for the peer review system developed was the personalised peer matching for improved learning from reviewing peers and also learn from feedback received from the peers. 9 students strongly agreed while 3 agreed to have learn from the feedback they received as shown in Figure 35.

During the peer review process, students were reviewing based on the assessment criteria and the reference document, all provided by the teacher while setting the assignment. After the completion of the review, they were asked how they found the assessment criteria to be helpful.

9 students strongly agree, while 3 out of 12 students agreed that the assessment criteria were useful in the review process as well as their learning benefits. The Figure 36 shows the assessment criteria usefulness in the post survey questionnaire.

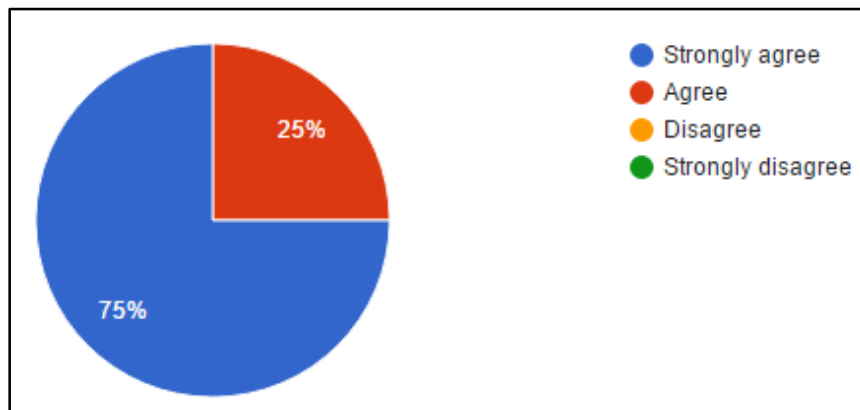


Figure 36: Post-survey on assessment criteria usefulness

Students were asked if they plan to complain about the feedback they received but no one agreed. This means that the feedbacks were satisfying based on the solution they had submitted, the reference code file provided by the teacher and the assessment criteria that were based on while reviewing.

After all, students were asked about the difficulty of the navigation through the peer review interfaces. 8 students found the system to be easy to use while 4 out of 12 students found it too easy to use. No student was confused about the interfaces according to their responses. In the development of peer review system, even though it was our priority to find a personalised peer matching algorithm, it was also among our main focus to make the interfaces easy to use by the end users. Figure 37 shows the post survey responses about the navigation through the interfaces.

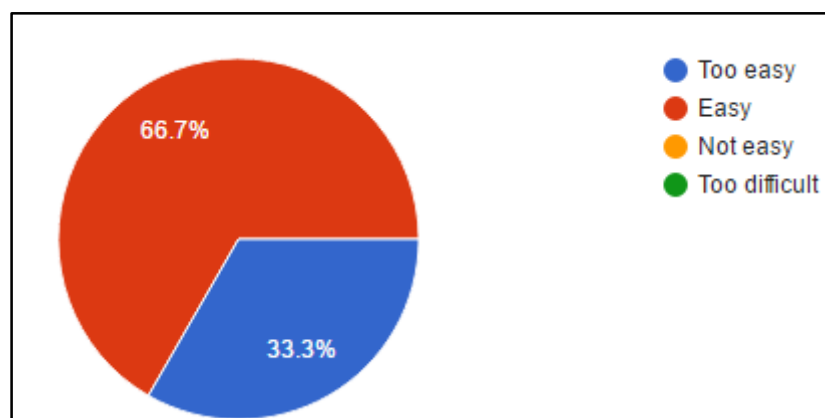


Figure 37: Post-survey responses for interface navigation

All students were satisfied by the system being tested and recommended it to be used in their studies most especially programming course as the core course of Information and Communication Technology without ignoring all other courses. The evaluation team composed of 12 students from first year computer science where 8 were male and 4 were female all aging from 18 to 30 years old. During the evaluation process, they were given 30 minutes of solving the question, 20 minutes for review process, 10 minutes for checking the feedback and 10 minutes to fill the post-survey questionnaire. Without any difficulty about the process, all students finished the task as expected and no student claimed about the time.

7. Discussion and limitations

7.1 Discussion

Personalized peer matching was found to be more helping to the students that are included in the peer review process. It fulfils the needs of a student as an individual with different needs from other students. During peer review process, students acquire knowledge from other students' work and also contribute to the learning of their fellow students. This also helps the teacher to minimize the marking overhead and the time used to give support to individual students that meet difficulties while solving assignment.

The peer review process is considered to be aiding the students by showing the most important features of the programming code, this is done through the effective assessment criteria. Teachers are responsible for the assessment criteria that are more crucial and relevant to the given assignment task and they should be concise on the weights for the assessments criteria. These criteria must be the topics covered by the assignment problem and this responds to the research question "What are the assessment criteria for reviewing the programming assignments which can be adapted in peer review for the introduction to programming course?" Giving weight to the assessment should be done in better manner considering the difficulty and the importance of the criterion in the learning. This must be done by the teacher while giving the criteria. Assessment criteria are the fundamental learning ground where teachers were also using them in the traditional marking of the assignments and we consider the teacher to provide the same weights as they were given traditionally.

Assessment criteria must be considered to be the key driver for effective peer review system as well as the efficient personalized peer matching algorithm. This is based on the fact that a specific student's weakness can be specified by the algorithm and s/he will be given a better student in the next review where s/he will get better feedback on the assignment review. These criteria are highlighted in order for the teacher to use them while giving assignments. The assessment criteria must be related to the assignment deliverables for the students to be marked on how they delivered the requested solution otherwise, students can be out of scope for the assignment requirements. The number of criteria per assignment must not be large as they will be tiresome to students and may cause un-seriousness in the review process and the feedback may not be effective.

For the peer review to be consistent, teachers provide a reference document that is used in the review process. This is a solution to the assignment given to be solved. We could ask ourselves

why the students should bother themselves while reviewing students work while they already have a solution from the teacher. The answer here comes back to the speciality of programming courses as critically said by teachers during the interview period, where all individuals have their own programming logic and coding style and techniques. Students will be able to look at different techniques used by the other students s/he is reviewing, as well as the techniques used by the teacher. This will not be necessary that the student's solution must be completely the same as the teacher's solution, but it must comply with the requirements of the assignment task. The reference code document must be there to give reference where necessary so that students should not invent what was not supposed to be in the solution.

Students are grouped in levels according to their performances; total criteria performance and individual criterion performances has created the motivating learning technique based on the students' needs. This is more helping to the students who need assistance to their solutions. This personalized grouping helped the students in the user testing assignment to know the benefits of using personalised peer matching. 12 students in the user testing confirmed these benefits when they obtained feedbacks on the solution parts they met difficulties to solve. These students' benefits from personalised peer matching responded to the research question; "Which is the optimal algorithm/approach that can be used to match peers that will be used in the peer review process?". Personalized peer matching was found the optimal approach to match peers based on their individual needs.

During the usability testing, the time used in the peer review was enough for the students to understand the problem requirements, solve the problem, submit the solution, make three reviews per each student and read the received feedback. Students used 1 hour 10 minutes to complete all the tasks whereas it was expected that they would take minimum 1 hour 30 minutes based on their availability. This depended on the requirements (reference document and assessment criteria) provided by the teacher in the assignment given, which make student get more help while giving feedback hence it becomes more important in students' learning process.

Students during the peer review process were allowed to give feedback and categorize the student based on the criterion being assessed. This was accomplished with reference to both reviewer's perspective and the reference document. No student is allowed to give grade to his fellow as said by the teachers. Students are allowed to view only the assignment task file when submitting the solution. After submission, s/he is able to view the reference document during the review process. This complies to the T-FLIP data structure where students are allowed to

only view assignment task and use the reference document in evaluating her/his submission. This responds to our last research question; “How can the approach concept be adapted to TFLIP features and data structure to improve the system usefulness?”.

In addition to students’ benefits, teachers are allowed to review all students whenever necessary with all requirements; reference document along with the assessment criteria s/he provided. This makes the work to be done in relatively shorter time compared to the time it could have taken while using traditional claim solving. Meanwhile, in the user testing no student claimed to have received irrelevant feedback.

According to the practice throughout this project, it is noticed that students’ learning can be improved if they are given proper feedback efficiently. The results and analysis of the surveys and interviews, provided sufficient basis to develop an assessment criterion. By forming an effective assessment criteria and introducing the personalized peer matching algorithm, we ensure that the students are motivated to participate in the peer review process by getting needed feedback about their work from different students. Not just that, but they review and give feedback to other peers by evaluating their work and contribute in the fixing of their weaknesses. This will definitely improve their programming skills, evaluating/judgement spirit, responsibility and eventually improve their learning abilities.

7.2 Limitations

During the development of this peer review system, we met a lot of challenges which led to delay and also not working as expected. The first challenge was the availability of teachers during the data collection time. Teachers were busy because it was in the middle of the semester and they were overloaded by teaching. This took time to get the required data and delayed the development phase. After the data collection from teachers, we had to collect data from students too. Access to students was more challenging because all students were busy with studies and no one wanted to give her/his time. It delayed that phase until we were advised to request teachers to send survey questionnaires to students for us. A survey questionnaire was sent to more than 150 students studying computer science if first year. 21 students out of 150 students responded to the survey. We used their responses with the responses from teachers to formulate the requirements.

The second challenge was met during the user testing period. Students were not interested to help us test the system and we had to offer incentives. Students were given free pizza and coffee or tea during the testing period but it did not help much. We use 12 students who accepted to help us in the testing. This number was not enough but we could not get more than that based

on the students' availability and schedules. We are not blaming them for not helping us because they had their scheduled.

The last challenge was the time for the project being not enough. It could have been better if the project period was two semesters. This could help us approach more teachers and also more students. We could have used the bigger samples during the data collection as well as during the user testing. This could give us more accurate perspectives from both groups (teachers and students). From these perspectives, we could obtain all the needed requirements to be included in the peer review system.

8. Conclusion and future work

8.1 Conclusion

Personalized learning has proved that a better learning is based on personal needs to acquire skills which can be used to overcome personal weaknesses. This helps learners to get more helped and improve their standards of learning. While personalizing the learning, many researches considered peer review to help in the learning within the same level of learners and it showed a good performance in different disciplines. Personalized peer matching is used in the distribution of needed reviewers to students and it helps them to receive help that can work as their guidance in the correction of their not satisfying solutions as well as improve their learning capabilities.

In this project, personalized peer matching was used for the purpose of improving the learning of programming courses. Personalized peer review was conducted and showed an improvement to the traditional peer review where students were obtaining feedbacks from assigned reviewers who are assigned either randomly or based on total performances. In this project, it is now clear that students need to receive feedbacks from different student levels depending on multiple factors. These factors being considered here are the total performance obtained by both the author of the assignment solution and the reviewer's performance, the assessment criteria which are the most leading factor being considered and finally the reference document provided by the teacher to guide the reviewers in the process.

A combination of all these factors made a great satisfying learning standard for the students who participated in the usability testing and this gives perspectives of the students who will be using the personalized peer review in their everyday learning style. Students are able to review their colleagues in transparent manner and it improves the quality of feedback received for learning purpose.

The system developed in this project will not definitely reduce the teacher's workload but it minimizes the workload. Teachers are able to make review only if a student request for it or if the teacher finds the given feedback not relevant nor efficient to the assignment solution provided by the author. Otherwise, teachers will be there for the control of the feedbacks given by the reviewers.

Therefore, a better learning of programming language courses can be significantly improved by applying a peer review process to minimize the teacher's workload. Not only for minimal teacher's workload, but also the better learning of different techniques from different students and also the perspectives of others for authors' solution and proposed solution from the colleagues.

We therefore conclude personalised peer matching combined with relevant and effective assessment criteria which are assigned to the programming assignment give a great advantage and a successful learning method for programming course.

8.2 Future work

The introduction of personalized peer matching is more important and advantageous to the researchers in the field of peer review to be improved for further use. As personalized peer review is missing in the literature, it an advantage to explore more on its use in all disciplines to improve learning based on learner's needs. As recommended by the students in the usability testing, we plan to explore more about assessment based peer review in different courses without limiting only to programming courses.

Furthermore, contextualised learning can be explored in details to improve needs based learning which can improve the learners' skills. This can be done through a context aware based system for learning. This system can help in the learning based on context awareness fundamentals.

References

- [1] Lombardi Marilyn M, “Authentic learning for the 21st century: An overview,” *Educause learning initiative 1*, vol. 1, no. 2007, pp. 1-12., 2007 .
- [2] Satu Alaoutinen, Kari Smolander, “Student Self-assessment in a Programming Course Using Bloom’s Revised Taxonomy,” in *Proceedings of the 15th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE*, Bilkent, Ankara, Turkey, 2010.
- [3] Yanqing Wang, Yaowen Liang, Luning Liu & Ying Liu, “A multi-peer assessment platform for programming language learning: considering group non-consensus and personal radicalness,” in *Interactive Learning Environments*, Long Beach, CA, USA, Taylor & Francis, 2015, pp. 1-20.
- [4] Ray Spier, “The history of the peer-review process,” *Trends in Biotechnology*, vol. XX, no. 8, p. 357–358, 2002.
- [5] Stephen Cole, Jonathan R. Cole, Gary A. Simon, “Chance and Consensus in Peer Review,” *Science, New Series*, vol. 214, no. 4523, pp. 881-886, 1981.
- [6] Rochelle F. Rubin, Traci Turner, “Student performance on and attitudes toward peer assessments on Advanced Pharmacy Practice Experience assignments,” *Currents in Pharmacy Teaching and Learning*, vol. 4, no. 2, p. 113–121, 2012.
- [7] Charlotte Murakami, Christopher Valvona, Daniel Broudy, “Turning apathy into activeness in oral communication classes: Regular self- and peer-assessment in a TBLT programme,” *System*, vol. 40, no. 3, p. 407–420, 2012.
- [8] Marjo van Zundert, Dominique Sluijsmans, Jeroen van Merriënboer, “Effective peer assessment processes: Research findings and future directions,” *Learning and Instruction*, vol. 20, no. 4, p. 270–279, 2010.
- [9] Mark Langan, Phil Wheeler, “Can students assess students effectively? Some insights into peer-assessment.,” *Learning and Teaching Unit*, vol. 2, no. 1, pp. ISSN 1477-1241, 2003.
- [10] Keith Topping, “Peer Assessment between Students in Colleges and Universities,” *American Educational Research Association , JSTOR*, vol. 68, no. 3, pp. 247-276, 1998.
- [11] Morten Goodwin (Project Coordinator), “T-FLIP: Testdrevet ferdighetsl ring i programmering,” 01 August 2015. [Online]. Available: <http://tflip.uia.no/>. [Accessed 20 March 2016].

- [12] Kari Smolander, "Student self-assessment in a programming course using bloom's revised taxonomy," in *Proceedings of the 15th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE 2010*, Bilkent, Ankara, Turkey, 2010.
- [13] Pierre Dillenbourg, "Collaborative Learning: Cognitive and Computational Approaches. Advances in Learning and Instruction Series," *Journal of Educational Computing Research*, vol. 22, no. 1, pp. 131-134, 2000.
- [14] Anuradha A. Gokhale, "Collaborative Learning Enhances Critical Thinking," *Journal of Technology Education*, vol. 7, no. 1, pp. 1-83, 1995.
- [15] James Atherton, "Learning and Teaching: Constructivist Theory," 10 February 2013. [Online]. Available: <http://www.learningandteaching.info/learning/constructivism.htm>. [Accessed 23 April 2016].
- [16] Lasse Lipponen, "Exploring foundations for computer-supported collaborative learning," in *CSCL '02 Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*, Boulder, CO, USA, 2002.
- [17] Koschmann Timothy , "Paradigm Shifts and Instructional Technology," in *In CSCL: Theory and Practice of an Emerging Paradigm*, Mahwah New Jersey, Lawrence Erlbaum Associates, 1996, p. 1–23.
- [18] Gerry Stahl, Timothy Koschmann, Dan Suthers, "Computer-supported collaborative learning: An historical perspective," in *Cambridge handbook of the learning sciences*, Cambridge, UK, Cambridge University Press, 2006, p. Cambridge handbook of the learning sciences.
- [19] Tosun, Nilgün; Suçsuz, Nurşen; Yiğit, Birol, "The effect of computer assisted and computer based teaching methods on computer course success and computer using attitudes of students," *The Turkish Online Journal of Educational Technology – TOJET*, vol. 5, no. 3, pp. 1-8, 2006.
- [20] J. Bull, "Computer-Assisted Assessment: Impact on Higher Education Institutions," *Educational Technology & Society*, vol. 2, no. 3, 1999.
- [21] Nils Ulltveit-Moe, Sigurd Assev, Terje Gjøsæter, Halvard Øysæd, "Streamlining Assessment Using a Knowledge Metric," 2016.
- [22] Kartinah Zen, D. N. F Awang Iskandar, Ongkir Linang, "Using Latent Semantic Analysis for automated grading programming assignments," in *2011 International Conference on Semantic Technology and Information Retrieval*, , Putrajaya, 2011.

- [23] M. v. Zundert, D. Sluijsmans and J. v. Merriënboer, "Effective peer assessment processes: Research findings and future directions," *Learning and Instruction*, vol. 20, no. 4, p. 270–279, 2010.
- [24] Chadi Aoun, "Peer-assessment and learning outcomes : product deficiency or process defectiveness?," in *proceedings of the 34th International Association for Educational Assessment (IAEA) Annual Conference*, Cambridge, UK, 2008.
- [25] Arthur Heather, "Student self-evaluations: How useful? How valid?," *International Journal for Nursing studies*, vol. 32, no. 3, p. 271–276, 1995.
- [26] Ghislain Maurice Norbert Isabwe, Frank Reichert, Martin Carlsen, Thomas Andr   Lian, "Using Assessment for Learning Mathematics with Mobile Tablet Based Solutions," *International Journal of Emerging Technologies in Learning*, vol. 9, no. 2, 2014.
- [27] Sheng-Chau Tseng, Chin-Chung Tsai, "On-line peer assessment and the role of the peer feedback: A study of high school computer course," *Computers & Education*, vol. 49, no. 4, pp. 1161-1174 , 2007.
- [28] Heng Luo, Anthony C. Robinson, Jae-Young Park, "Peer Grading In A MOOC: Reliability, Validity, And Perceived Effects," *ONLINE LEARNING JOURNA*, vol. 18, no. 2, pp. 1-14, 2014.
- [29] K. S. Satu Alaoutinen, "Student Self-assessment in a Programming Course Using Bloom's Revised Taxonomy," in *Proceedings of the 15th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE*, Bilkent, Ankara, Turkey, 2010.
- [30] David R. Krathwohl, "A Revision of Bloom's Taxonomy: An Overview," *Theory Into Practice*, vol. 41, no. 4, pp. 212-218, 2010.
- [31] D. M. Morrisson, "PRISM: A Peer Review Managements System for Programming Assignments," *Issues in Information Systems*, vol. V, no. 2, pp. 626-632, 2004.
- [32] Wang Yanqing, Li Hang, Sun Yanan, Jiang Yu, Yu Jie, "Learning outcomes of programming language courses based on peer code review model," in *The 6th International Conference on Computer Science & Education (ICCSE 2011)*, Virgo, Singapore, 2011.
- [33] Saikrishna Sripada, Y. Raghu Reddy, Ashish Sureka, "In Support of Peer Code Review and Inspection in an Undergraduate Software Engineering Course," in *2015 IEEE 28th Conference on Software Engineering Education and Training*, Florence, 2015.
- [34] PKP, "Unit 1: The Different Types of Peer Review," 10 July 2015. [Online]. Available: <http://pkpschool.sfu.ca/becoming-an-editor/module-7/unit-1-the-different-types-of-peer-review/>. [Accessed 2 February 2016].

- [35] Morrisson, Dr. Mike, "PRISM: A Peer Review Managements System for Programming Assignments," *Issues in Information Systems*, vol. V, no. 2, pp. 626-632, 2004.
- [36] algorithms.net, "Fisher-Yates shuffle," 2015. [Online]. Available: <http://www.programming-algorithms.net/article/43676/Fisher-Yates-shuffle>. [Accessed 20 March 2016].
- [37] Michael McCracken, Vicki Almstrum, Danny Diaz, Mark Guzdial, Dianne Hagan, Yifat Ben-David Kolikant, Cary Laxer, Lynda Thomas, Ian Utting, Tadeusz Wilusz, "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students," *Newsletter ACM SIGCSE Bulletin*, vol. 33, no. 4, pp. 125-180, 2001.
- [38] Christine Bruce, Lawrence Buckingham, John Hynd, Camille McMahon, Mike Roggenkamp, Ian Stoodley, "Ways of Experiencing the Act of Learning to Program: A Phenomenographic Study of Introductory Programming Students at University," *Journal of Information Technology Education*, vol. III, pp. 143-160, 2004.
- [39] Dehnadi Saeed, "Testing programming aptitude," in *Proceedings of the 18th Annual Workshop of the Psychology of Programming Interest Group*, Brighton, England, 2006.
- [40] Marilyn Ford and Sven Venem, "Assessing the Success of an Introductory Programming Course," *Journal of Information Technology Education*, vol. 9, pp. 133-145, 2010.
- [41] Maurizio Morisio, Marco Torchiano, Giovanni Argentieri, "Assessing Quantitatively a Programming Course," in *Proceedings of the 10th International Symposium on Software Metrics (METRICS'04) 1530-1435/04 \$ 20.00 IEEE*, Washington, DC, USA, 2004.
- [42] Kirsti Ala-Mutka, "Problems in learning and teaching programming," 2010. [Online]. Available: https://www.cs.tut.fi/~edge/literature_study.pdf. [Accessed 20 March 2016].
- [43] John F. Pane, Brad A. Myers, "Usability Issues in the Design of Novice Programming also appear as: Human-Computer Interaction Institute Technical Report CMU-HCII-96-101," School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1996.
- [44] Jason Cohen, "IBM DeveloperWorks: 11 proven practices for more effective, efficient peer code review," 25 January 2011. [Online]. Available: <http://www.ibm.com/developerworks/rational/library/11-proven-practices-for-peer-review/>. [Accessed 10 February 2016].
- [45] A. L. M. o. S. & T. Operations, Interviewee, *E-mail communication*. [Interview]. 21 April 2016.
- [46] ISO 9241-210, 2010, Ergonomics of human-system interaction.

- [47] James & Suzanne Robertson, “Volere Requirements Specification Template,” 2014. [Online]. Available: <http://www.volere.co.uk/template.htm>. [Accessed 19 February 2016].
- [48] Marjo van Zundert, Dominique Sluijsmans, Jeroen van Merriënboer, “Effective peer assessment processes: Research findings and future directions,” *Learning and Instruction*, vol. 20, no. 4, p. 270–279, 2010.
- [49] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, Charles E. Youman , “Role-Based Access Control Models,” *IEEE Computer*, vol. 29, no. 2, pp. 38-47, 1996.
- [50] Gene Cooperman, Larry Finkelstein, “Combinatorial Tools for Computational Group Theory,” *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 11, no. 2, pp. 53-86, 1993.
- [51] Gene Cooperman, Larry Finkelstein, “A Random Base Change Algorithm for Permutation Groups,” *Journal of Symbolic Computation*, vol. 17, no. 6, pp. 513-528, 1994.
- [52] László Babai, “Bounded round interactive proofs in finite groups,” *SIAM Journal on Discrete Mathematics*, vol. 5, no. 1, pp. 88 - 111 , 1992.
- [53] László Babai, Eugene M. Luks, Ákos Seress, “Fast Management of Permutation Groups,” in *The 29th IEEE Symposium on Foundations of Computer Science*, White Plains, 1988.
- [54] Babai László, Cooperman Gene, Finkelstein Larry, Luks Eugene M, Seress Ákos, “Fast Monte Carlo Algorithms for Permutation Groups,” *Journal of Computer and System Sciences*, vol. 50, no. 2, p. 296–308, 1995.
- [55] Deborah J. Mayhew, Principles and guidelines in software user interface design, Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992 .
- [56] M. Johnson, “Personalised learning,” *New Economy*, vol. 11, no. 4, pp. 224-228, 2004.
- [57] Howard Gardners, “Class divisions: Who benefits from the ‘personalised learning’ strategy of dividing school pupils into subsets.,” 2008. [Online]. Available: <http://www.infed.org/mobi/howard-gardner-multiple-intelligences-and-education/>. [Accessed 20 April 2016].
- [58] T. L. Good and J. E. Brophy, Educational psychology: A realistic approach, 4 ed., New York: Longman/Addison Wesley Longman, 1990.
- [59] D. Benyon, Designing Interactive Systems: A comprehensive guide to HCI and interaction design, Second ed., Essex: Pearson Education Limited, 2010.
- [60] William Hart-Davidson, Michael McLeod, Christopher Klerkx, Michael Wojcik, “Introducing Students to Peer Review of Writting,” in *SIGDOC '10 Proceedings of the*

- [61] Aika Qazi, Karim Bux Shah Syed, Ram Gopal Raj, Erik Cambria, Muhammad Tahir, Daniyal Alghazzawi, “A concept-level approach to the analysis of online review helpfulness,” *Computers in Human Behavior*, vol. 58, no. C, pp. 75-81 , 2016.
- [62] Andrew Dillon, “Usability evaluation,” in *W. Karwowski (ed.) Encyclopedia of Human Factors and Ergonomics*, London, Taylor and Francis, 2001.
- [63] Jakob Nielsen, “How to Conduct a Heuristic Evaluation,” 1 January 1995. [Online]. Available: <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>. [Accessed 25 April 2016].
- [64] Cathleen Wharton , John Rieman, Clayton Lewis, Peter Polson, “The cognitive walkthrough method: a practitioner's guide,” in *Usability inspection methods*, New York, NY, USA, John Wiley & Sons, Inc, 1994 , pp. 105 - 140 .

Appendices

Appendix A: Assignment task and solution submitted to T-FLIP

I denne oppgaven skal dere lage et enkelt program for å fly i stjernene. Se et eksempel på programmet i figur 1 og figur 2.



Figur 1: Skjermdump av programmet



Figur 2: Skjermdump av programmet med frivillig

1. Lag en klasse **Oblig7** som arver fra **JFrame**. Denne klassen må ha navnet **Oblig7**.
2. Gjør slik at den setter seg selv til synlig.
3. Lag en **Runner**-klasse med en **main**-metode. Denne skal kun gjøre en ting: Lage et objekt av **Oblig7**-klassen. Denne klassen må ha navnet **Oblig7Runner**.
4. I **Oblig7**-klassen, legg til et en variabel som skal holde på punktene. Variabelen skal i senere oppgave ta imot et ukjent antall punkter og være tilgjengelig fra flere metoder.

5. Legg til minst 4 objekter av **Point** i klassevariabelen fra oppgave 4. Du velger selv x- og y-verdier.
6. Kopier inn følgende metode i klassen din. Inne i denne metoden skal du gjøre alt av opptegninger i oppgavene.

```
1 @Override
2 public void paint(Graphics g){
3     super.paint(g);
4 }
```

7. Inne i **paint**-metoden skal du hente ut det første punktet (fra klassevariabelen i oppgave 4) og bruke dette til å tegne en sirkel. For å tegne en sirkel når du vet x- og y-verdiene kan du gjøre som følger:

```
1 g.fillOval(XKOORDINAT,YKOORDINAT,BREDDE,HØYDE);
```

8. Inne i **paint**-metoden skal du iterere igjennom alle punktene og tegne en sirkel for hvert punkt.
9. Gjør en endring slik at du istedenfor å kun legge til 4 punkter (fra oppgave 5), legger du til minst 100. Hint: For å lage et tilfeldig tall kan man bruke et objekt av **Random**, som følger.

```
1 Random rg = new Random();
2 int tall = rg.nextInt(1000); //Tilfeldig tall mellom 0 og 1000.
```

10. Oppdater **paint**-metoden slik at du itererer igjennom punktene en gang til. Denne gangen skal du tegne en sirkel litt til venstre for selve punktet. For eksempel, hvis et punkt har x-koordinat 100, kan du tegne opp en ny sirkel med x-koordinat 99.
11. **Frivillig:** Legg til en timer hele tiden flytter punktene mot venstre. Hint: For å legge til en timer må du ha et objekt av **Timer** som følger:

```
1 Timer t = new Timer(this,1000);
2 t.start();
```

og klassen din må ha "Implements **ActionListener**" (se <https://docs.oracle.com/javase/tutorial/uiswing/events/actionlistener.html>). I **ActionPerformed** må du ha med:

Figure 38: Assignment task

```

package Tests; import static org.junit.Assert.*;

import java.io.ByteArrayOutputStream; import java.io.PrintStream; import TestTools.Tools;

import org.junit.After; import org.junit.Before; import org.junit.Test; import oblig1.Oblig1;

public class Oblig1Test {

    ByteArrayOutputStream c = new ByteArrayOutputStream();

    @Before

    public void setUpStreams(){

        System.setOut(new PrintStream(c));

        Tools.JOptionPane.addRule("Hva heter du?", "Morten");

    }

    @After

    public void cleanUpStreams(){

        System.setOut(null);

    }

    @Test(timeout=10000)

    public void Assignment1() {

        Oblig1.main(new String[]{});

    }

    @Test(timeout=10000)

    public void Assignment2() {

        Oblig1.main(new String[]{});

        assertTrue(c.toString().toLowerCase().contains("dette er mitt frste
program")); }

```

```

@Test(timeout=10000)

    public void Assignment3() {

        Oblig1.main(new String[]{});

        assertTrue(c.toString().toLowerCase().contains("dette er mitt\r\nfrste
program") || c.toString().toLowerCase().contains("dette er mitt\nfrste program"));

    }

    @Test(timeout=10000)

    public void Assignment4() {

        Oblig1.main(new String[]{});

        assertTrue(c.toString().toLowerCase().contains("messagedialog:dette er mitt
frste program"));

    }

    @Test(timeout=10000)

    public void Assignment5() {

        Oblig1.main(new String[]{});

        String data = c.toString().toLowerCase();

        if(!(data.contains("inputdialog:hva heter du") || data.contains("inputdialog:kva heiter
du"))){

            fail("Fant ikke teksten:hva heter du");

        }

    }

}

```

Figure 39: Assignment solution

Appendix B: System interfaces

Courses **Reviews** **Profile** **Log out** jef :Teacher

DAT001

DAT001 Basic Programming

This course is the introduction to programming language where we will be using Java as the programming tool. It contains 10 lectures and several assignments. Hand-in must be done before deadline

Figure 40: An added course

Courses **Reviews** **Profile** **Log out** jef :Teacher

DAT001

DAT001 Basic Programming

This course is the introduction to programming language where we will be using Java as the programming tool. It contains 10 lectures and several assignments. Hand-in must be done before deadline

| Assignment Name | Open date | Close date | Review Start | Review end | | | | |
|-----------------------------|---------------------|---------------------|---------------------|---------------------|------|--------|----------|--------------|
| Introduction to variables | 12/03/2016 00:00:00 | 08/04/2016 00:00:00 | 11/04/2016 00:00:00 | 15/04/2016 00:00:00 | Open | Submit | Criteria | Start Review |
| If statements | 06/04/2016 00:00:00 | 10/04/2016 00:00:00 | 11/04/2016 00:00:00 | 13/04/2016 00:00:00 | Open | Submit | Criteria | Start Review |
| Number Method: Prime number | 19/04/2016 00:00:00 | 20/04/2016 00:00:00 | 21/04/2016 00:00:00 | 22/04/2016 00:00:00 | Open | Submit | Criteria | Start Review |

Figure 41: Course related assignments

New Criterion for: Number Method: Prime number

Criterion

Weight

Figure 42: Adding a criterion to assignment

Courses **Reviews** **Profile** **Log out** stud1 :Student

DAT001

DAT001 Basic Programming

This course is the introduction to programming language where we will be using Java as the programming tool. It contains 10 lectures and several assignments. Hand-in must be done before deadline

| Assignment Name | Open date | Close date | Review Start | Review end | | |
|-----------------------------|---------------------|---------------------|---------------------|---------------------|------|--------|
| Introduction to variables | 12/03/2016 00:00:00 | 08/04/2016 00:00:00 | 11/04/2016 00:00:00 | 15/04/2016 00:00:00 | Open | Submit |
| If statements | 06/04/2016 00:00:00 | 10/04/2016 00:00:00 | 11/04/2016 00:00:00 | 13/04/2016 00:00:00 | Open | Submit |
| Number Method: Prime number | 19/04/2016 00:00:00 | 20/04/2016 00:00:00 | 21/04/2016 00:00:00 | 22/04/2016 00:00:00 | Open | Submit |

Number Method: Prime number

Enter a number and display the prime numbers below that number

Attach Answer file: No file chosen

Figure 43: Submitting assignment solution

Introduction to Programming Assignment

Prime Numbers:

Question: Write a program to find all prime number up to a given number? You will be prompted to enter the number you want to check the prime numbers less than that number you enter. You will display all the prime numbers.

Figure 44: Assignment sample to peer review system

```
import java.util.Scanner;

public class PrimeNumbers {

    public static void main(String[] args) {

        System.out.println("Enter the number till which prime number to be printed: "); // Message to tell the user to enter the number
        int limit = new Scanner(System.in).nextInt(); // input way for the number

        //printing prime numbers till the limit ( 1 to limit)
        System.out.println("Printing prime number from 1 to " + limit);
        for(int number = 2; number<=limit; number++){ //bounding all the numbers till limit
            //print prime numbers only
            if(isPrime(number)){ // isPrime is a method to return prime numbers only
                System.out.println(number); // display primes
            }
        }

    }

    public static boolean isPrime(int number){ //A method to select prime numbers
        for(int i=2; i<number; i++){ // testing if it's not divisible by any number less than itself
            if(number%i == 0){ //test
                return false; //number is divisible so its not prime
            }
        }
        return true; //number is prime now
    }

}
```

Figure 45: Submitted reference solution

Appendix C: Post-user testing survey for peer review system

After completion of peer review process, you are able to tell about the experience you met with the system either bad or good. Here below are few questions about the system.

How did you find the peer review helpfulness?

- ☐ Not helpful
- ☐ Helpful
- ☐ More helpful

How was the review process while giving feedback to other students?

- ☐ Too easy
- ☐ Easy
- ☐ Not easy
- ☐ Too difficult

Did you learn anything from reviewing other students?

- ☐ Strongly agree
- ☐ Agree
- ☐ Disagree
- ☐ Strongly disagree

Were the reviewers distributed in a better way?

- ☐ Strongly agree
- ☐ Agree
- ☐ Disagree
- ☐ Strongly disagree

Did you learn anything from feedback received from other students?

- ☐ Strongly agree
- ☐ Agree
- ☐ Disagree
- ☐ Strongly disagree

Do you plan to make a complaint about the feedback?

- ☐ Yes
- ☐ No

How the was navigation of system interfaces?

- ☐ Too easy
- ☐ Easy
- ☐ Not easy
- ☐ Too difficult

Appendix D: Peer matching code

```
protected void StartReview(object sender, EventArgs e)
{
    int id = int.Parse((sender as Button).CommandArgument);
    AssId = id;
    int PrevAss=id-1;
    SqlDataAdapter da = new SqlDataAdapter();
    DataTable ds = new DataTable();
    conn.Open();
    SqlCommand cm = new SqlCommand();
    cm.CommandText = "select distinct AnswerId,UserId from Answers where
AssignmentId=" + PrevAss + "";
    cm.Connection = conn;
    da.SelectCommand = cm;
    da.Fill(ds);
    conn.Close();
    txtlev.Text = " prev students " + (ds.Rows.Count).ToString();
    if (ds.Rows.Count == 0)
    {
        RandomAssignment();
    }
    else
    {

```



```

try{
    string constr =
ConfigurationManager.ConnectionStrings["PeerReviewConnectionString"].ConnectionString
;

    using (SqlConnection con = new SqlConnection(constr))
    {
        using (SqlCommand cmd = new SqlCommand())
        {
            cmd.CommandText = "select count(AnswerId) as Num from Answers
where AssignmentId=" + AssId + """;
            cmd.Connection = con;
            adp.SelectCommand = cmd;
            adp.Fill(dt);
            con.Open();
            N = Convert.ToInt32(dt.Rows[0]["Num"].ToString());
            con.Close();
            using (SqlCommand cmd0 = new SqlCommand())
            {
                cmd0.CommandText = "select UserId from Answers where
AssignmentId=" + id + """;
                cmd0.Connection = con;
                adp0.SelectCommand = cmd0;
                adp0.Fill(dt0);
                con.Open();
                for (int i = 0; i < dt0.Rows.Count; i++)
                {
                    students[i] = Convert.ToInt32(dt0.Rows[i]["UserId"].ToString());
                    con.Close();
                }
                using (SqlCommand cmd = new SqlCommand())
                {
                    cmd.CommandText = "select top " + (N / 3) + " UserId,Performance
from Reviews where AssignmentId=" + PrevAss + " order by Performance";
                    cmd.Connection = con;
                    adp1.SelectCommand = cmd;
                    adp1.Fill(dt1);
                    con.Open();
                    for (int i = 0; i < dt1.Rows.Count; i++)
                    {
                        low[i] = Convert.ToInt32(dt1.Rows[i]["UserId"].ToString());
                    }
                    con.Close();
                }
                using (SqlCommand cmd1 = new SqlCommand())
                {
                    cmd1.CommandText = "select * from (select Row_Number() over
(order by Performance) as RowIndex, * from Reviews) as Sub Where Sub.RowIndex >= " +
((N / 3) + 1) + " and Sub.RowIndex <=" + (2 * (N / 3)) + " and AssignmentId=" + PrevAss +
""";
                    cmd1.Connection = con;
                    adp2.SelectCommand = cmd1;
                    adp2.Fill(dt2);

```

```

        con.Open();
        for (int i = 0; i < dt2.Rows.Count; i++)
        {
            medium[i] = Convert.ToInt32(dt2.Rows[i]["UserId"].ToString());
        }
        con.Close();
    }
    using (SqlCommand cmd2 = new SqlCommand())
    {
        cmd2.CommandText = "select * from (select Row_Number() over
(order by Performance) as RowIndex, * from Reviews) as Sub Where Sub.RowIndex >= " +
(2 * (N / 3) + 1) + " and Sub.RowIndex <=" + N + "and AssignmentId=" + PrevAss + """;
        cmd2.Connection = con;
        adp3.SelectCommand = cmd2;
        adp3.Fill(dt3);
        con.Open();
        for (int i = 0; i < dt3.Rows.Count; i++)
        {
            high[i] = Convert.ToInt32(dt3.Rows[i]["UserId"].ToString());
            con.Close();
        }
    }
    int numToRemove = 0;
    low = low.Where(val => val != numToRemove).ToArray();
    medium = medium.Where(val => val != numToRemove).ToArray();
    high = high.Where(val => val != numToRemove).ToArray();
    int Re = 0;
    students = students.Where(val => val != Re).ToArray();
    int[] reviews1 = new int[N]; // total no of reviewers equal to the students' no
    int[] reviews2 = new int[N];
    int[] reviews3 = new int[N];
    for (int i = 0; i < low.Length; i++)
    {
        txtlev.Text += "\n low: " + low[i].ToString();
        txtlev.Text += " medium: " + medium[i].ToString();
        txtlev.Text += " high: " + high[i].ToString();
    }
    for (int i = 0; i < students.Length; i++)
    {
        int st;
        FoundL:
        st = ra.Next(0, low.Length);
        countlow[st] = countlow[st] + 1;
        if (countlow[st] <= 3) //&& low[st]!=students[i] to remove self review
        {
            reviews1[i] = low[st];
        } else
        { goto FoundL; } //Get another random without assigning reviewer
    }
    for (int i = 0; i < students.Length; i++)
    {

```

```

    int st;
FoundM:
    st = ra.Next(0, medium.Length);
    countmedium[st] = countmedium[st] + 1;
    if (countmedium[st] <= 3) //&& medium[st] != students[i] to be used to
remove self review
    {
        reviews2[i] = medium[st];
    }
    else
    { goto FoundM; } //Get another random without assigning reviewer
}
for (int i = 0; i < students.Length; i++)
{
    int st;

    // low = low.Where(val => val != rem1).ToArray();
FoundH:
    st = ra.Next(0, high.Length);
    counthigh[st] = counthigh[st] + 1;
    if (counthigh[st] <= 3) //&& medium[st] != students[i] to be used to remove
self review
    {
        reviews3[i] = high[st];
    }
    else
    { goto FoundH; } // Get another random without assigning reviewer
}
for (int i = 0; i < students.Length; i++)
{
    int temp;
    if (students[i] == reviews1[i] && i == 0)
    {
        temp = reviews1[i];
        reviews1[i] = reviews1[i + 2];
        reviews1[i + 2] = temp;
    }
    else if (students[i] == reviews1[i] && i == students.Length - 1)
    {
        temp = reviews1[i];
        reviews1[i] = reviews1[i - 2];
        reviews1[i - 2] = temp;
    }
    else if (students[i] == reviews1[i] && i != 0 && i != students.Length - 1)
    {
        temp = reviews1[i];
        reviews1[i] = reviews1[i + 1];
        reviews1[i + 1] = temp;
    }
    if (students[i] == reviews1[1])
        // goto Start;
        //***** check reviewer 2

```

[illegible]

```

        reviews1[i] = reviews1[i - 2];
        reviews1[i - 2] = temp;
    }
    else if (students[i] == reviews1[i] && i != 0 && i != students.Length - 1)
    {
        temp = reviews1[i];
        reviews1[i] = reviews1[i + 1];
        reviews1[i + 1] = temp;
    }
    if (students[i] == reviews1[1])
        // goto Start;
        //***** check reviewer 2
        if (students[i] == reviews2[i] && i == 0)
        {
            temp = reviews2[i];
            reviews2[i] = reviews2[i + 2];
            reviews2[i + 2] = temp;
        }
        else if (students[i] == reviews2[i] && i == students.Length - 1)
        {
            temp = reviews2[i];
            reviews2[i] = reviews2[i - 2];
            reviews2[i - 2] = temp;
        }
        else if (students[i] == reviews2[i] && i != 0 && i != students.Length - 1)
        {
            temp = reviews2[i];
            reviews2[i] = reviews2[i + 2];
            reviews2[i + 2] = temp;
        }
    }
    //***** check reviewer 3
    if (students[i] == reviews3[i] && i == 0)
    {
        temp = reviews3[i];
        reviews3[i] = reviews3[i + 1];
        reviews3[i + 1] = temp;
    }
    else if (students[i] == reviews3[i] && i == students.Length - 1)
    {
        temp = reviews3[i];
        reviews3[i] = reviews3[i - 1];
        reviews3[i - 1] = temp;
    }
    else if (students[i] == reviews3[i] && i != 0 && i != students.Length - 1)
    {
        temp = reviews3[i];
        reviews3[i] = reviews3[i + 1];
        reviews3[i + 1] = temp;
    }
}

DateTime today = new DateTime();
today = DateTime.Now;

```

```

        conn.Open();
        for (int i = 0; i < students.Length; i++)
        {
            txtlev.Text += "\n " + students[i] + " last has : " + reviews1[i].ToString()
+ reviews2[i].ToString() + reviews3[i].ToString();
        }
        for (int i = 0; i < students.Length; i++)
        {
            String insertuser = "insert into Reviewers (StudentId, Reviewer1,
Reviewer2, Reviewer3, CreationDate, AssignmentId) values (@std, @rev1, @rev2, @rev3,
@cdate, @assid)";
            SqlCommand com = new SqlCommand(insertuser, conn);
            com.Parameters.AddWithValue("@std", students[i]);
            com.Parameters.AddWithValue("@rev1", reviews1[i]);
            com.Parameters.AddWithValue("@rev2", reviews2[i]);
            com.Parameters.AddWithValue("@rev3", reviews3[i]);
            com.Parameters.AddWithValue("@cdate", today);
            com.Parameters.AddWithValue("@assid", AssId);
            com.ExecuteNonQuery();
            txtlev.Text += "\n " + students[i] + " inside has : " +
reviews1[i].ToString() + reviews2[i].ToString() + reviews3[i].ToString();
        }
        conn.Close();
        Page.ClientScript.RegisterStartupScript(this.GetType(), "Scripts", "<script>
alert('Reviewer successfully assigned based on last assignment') </script>");
    }
    catch (Exception ex)
    {
        ScriptManager.RegisterStartupScript(this, this.GetType(), "Message",
"alert('Error occurred : " + ex.Message.ToString() + "');", true);
    }
    finally
    {
        {
            dt.Clear();
            dt.Dispose();
            adp.Dispose();
        }
    }
}

private void RandomAssignment()
{
    int NumOfSt;
    DataTable d = new DataTable();
    SqlDataAdapter ad = new SqlDataAdapter();
    int[] studs = new int[200];
    int[] rev1 = new int[200];
    int[] rev2 = new int[200];
    int[] rev3 = new int[200];
    string constr =
ConfigurationManager.ConnectionStrings["PeerReviewConnectionString"].ConnectionString
;
    SqlConnection con = new SqlConnection(constr);

```

```

SqlCommand cm = new SqlCommand();
cm.CommandText = "select distinct AnswerId,UserId from Answers where
AssignmentId="+ AssId +"";
cm.Connection = con;
ad.SelectCommand = cm;
ad.Fill(d);
con.Open();
NumOfSt = d.Rows.Count;
for (int i = 0; i < d.Rows.Count; i++)
{
    studs[i] = Convert.ToInt32(d.Rows[i]["UserId"].ToString());
    if (i == NumOfSt - 3)
    {
        rev1[i] = Convert.ToInt32(d.Rows[i + 1]["UserId"].ToString());
        rev2[i] = Convert.ToInt32(d.Rows[i+2]["UserId"].ToString());
        rev3[i] = Convert.ToInt32(d.Rows[0]["UserId"].ToString());
    }
    else if (i == NumOfSt - 2)
    {
        rev1[i] = Convert.ToInt32(d.Rows[i+1]["UserId"].ToString());
        rev2[i] = Convert.ToInt32(d.Rows[0]["UserId"].ToString());
        rev3[i] = Convert.ToInt32(d.Rows[1]["UserId"].ToString());
    }
    else if (i == NumOfSt - 1)
    {
        rev1[i] = Convert.ToInt32(d.Rows[0]["UserId"].ToString());
        rev2[i] = Convert.ToInt32(d.Rows[1]["UserId"].ToString());
        rev3[i] = Convert.ToInt32(d.Rows[2]["UserId"].ToString());
    }
    else
    {
        rev1[i] = Convert.ToInt32(d.Rows[i+1]["UserId"].ToString());
        rev2[i] = Convert.ToInt32(d.Rows[i+2]["UserId"].ToString());
        rev3[i] = Convert.ToInt32(d.Rows[i+3]["UserId"].ToString());
    }
}
con.Close();
try
{
    DateTime today = new DateTime();
    today = DateTime.Now;
    conn.Open();
    int numToRemove = 0;
    studs = studs.Where(val => val != numToRemove).ToArray();
    rev1 = rev1.Where(val => val != numToRemove).ToArray();
    rev2 = rev2.Where(val => val != numToRemove).ToArray();
    rev3 = rev3.Where(val => val != numToRemove).ToArray();
    for (int i = 0; i < d.Rows.Count; i++)
    {
        String insertuser = "insert into Reviewers (StudentId, Reviewer1, Reviewer2,
Reviewer3, CreationDate, AssignmentId) values (@std, @rev1, @rev2, @rev3, @cdate,
@assid)";

```

```

        SqlCommand com = new SqlCommand(insertuser, conn);
        com.Parameters.AddWithValue("@std", studs[i]);
        com.Parameters.AddWithValue("@rev1", rev1[i]);
        com.Parameters.AddWithValue("@rev2", rev2[i]);
        com.Parameters.AddWithValue("@rev3", rev3[i]);
        com.Parameters.AddWithValue("@cdate", today);
        com.Parameters.AddWithValue("@assid", AssId);

        com.ExecuteNonQuery();
    }
    conn.Close();
    Page.ClientScript.RegisterStartupScript(this.GetType(), "Scripts", "<script>
alert('Done!! Reviewer successfully assigned') </script>");

}
catch (Exception ex)
{
    Response.Write("Error:" + ex.ToString());
}
}
private void Reviewers()
{
    int numToRemove = 0;

    students = students.Where(val => val != numToRemove).ToArray();

    int[] reviews1 = new int[students.Length]; // total no of reviewers equal to the students'
no
    int[] reviews2 = new int[students.Length];
    int[] reviews3 = new int[students.Length];

    txtlev.Text += " Num: " + (students.Length).ToString();
    for (int i = 0; i < low.Length; i++)
    {
        txtlev.Text += "\n low: " + low[i].ToString();
        txtlev.Text += " medium: " + medium[i].ToString();
        txtlev.Text += " high: " + high[i].ToString();
    }
    for (int i = 0; i < students.Length; i++)
    {
        int st;
        FoundL:
        st = ra.Next(0, low.Length);
        countlow[st] = countlow[st] + 1;
        if (countlow[st] <= 3)
        {
            reviews1[i] = low[st];
        }
        else
        { goto FoundL; } //Get another random without assigning reviewer
    }
    for (int i = 0; i < reviews1.Length; i++)

```



```

    {
        txtlev.Text += " rev1: " + reviews1[i].ToString();
    }
    for (int i = 0; i < students.Length; i++)
    {
        int st;
        FoundM:
        st = ra.Next(0, medium.Length);
        countmedium[st] = countmedium[st] + 1;

        if (countmedium[st] <= 3)
        {
            reviews2[i] = medium[st];
        }
        else
        { goto FoundM; }
    }
    for (int i = 0; i < students.Length; i++)
    {
        int st;
        FoundH:
        st = ra.Next(0, high.Length);
        counthigh[st] = counthigh[st] + 1;

        if (counthigh[st] <= 3)
            reviews3[i] = high[st];
        }
        else
        { goto FoundH; } // Get another random without assigning reviewer
    }

    for (int i = 0; i < students.Length; i++)
    {
        int temp;
        //-----check reviewer 1
        //Start:
        if (students[i] == reviews1[i] && i == 0)
        {
            temp = reviews1[i];
            reviews1[i] = reviews1[i + 2];
            reviews1[i + 2] = temp;
        }
        else if (students[i] == reviews1[i] && i == students.Length - 1)
        {
            temp = reviews1[i];
            reviews1[i] = reviews1[i - 2];
            reviews1[i - 2] = temp;
        }
        else if (students[i] == reviews1[i] && i != 0 && i != students.Length - 1)
        {
            temp = reviews1[i];
            reviews1[i] = reviews1[i + 1];

```

```

        reviews1[i + 1] = temp;
    }
    if (students[i] == reviews1[1])
        //***** check reviewer 2
        if (students[i] == reviews2[i] && i == 0)
        {
            temp = reviews2[i];
            reviews2[i] = reviews2[i + 2];
            reviews2[i + 2] = temp;
        }
        else if (students[i] == reviews2[i] && i == students.Length - 1)
        {
            temp = reviews2[i];
            reviews2[i] = reviews2[i - 2];
            reviews2[i - 2] = temp;
        }
        else if (students[i] == reviews2[i] && i != 0 && i != students.Length - 1)
        {
            temp = reviews2[i];
            reviews2[i] = reviews2[i + 2];
            reviews2[i + 2] = temp;
        }
    }
    if (students[i] == reviews3[i] && i == 0)
    {
        temp = reviews3[i];
        reviews3[i] = reviews3[i + 1];
        reviews3[i + 1] = temp;
    }
    else if (students[i] == reviews3[i] && i == students.Length - 1)
    {
        temp = reviews3[i];
        reviews3[i] = reviews3[i - 1];
        reviews3[i - 1] = temp;
    }
    else if (students[i] == reviews3[i] && i != 0 && i != students.Length - 1)
    {
        temp = reviews3[i];
        reviews3[i] = reviews3[i + 1];
        reviews3[i + 1] = temp;
    }
    //----- the second time
}
for (int i = 0; i < students.Length; i++)
{
    int temp;
    //-----check reviewer 1
    //Start:
    if (students[i] == reviews1[i] && i == 0)
    {
        temp = reviews1[i];
        reviews1[i] = reviews1[i + 2];
        reviews1[i + 2] = temp;
    }
}

```

```

}
else if (students[i] == reviews1[i] && i == students.Length - 1)
{
    temp = reviews1[i];
    reviews1[i] = reviews1[i - 2];
    reviews1[i - 2] = temp;
}
else if (students[i] == reviews1[i] && i != 0 && i != students.Length - 1)
{
    temp = reviews1[i];
    reviews1[i] = reviews1[i + 1];
    reviews1[i + 1] = temp;
}
if (students[i] == reviews1[1])
    // goto Start;
    //***** check reviewer 2
    if (students[i] == reviews2[i] && i == 0)
    {
        temp = reviews2[i];
        reviews2[i] = reviews2[i + 2];
        reviews2[i + 2] = temp;
    }
    else if (students[i] == reviews2[i] && i == students.Length - 1)
    {
        temp = reviews2[i];
        reviews2[i] = reviews2[i - 2];
        reviews2[i - 2] = temp;
    }
    else if (students[i] == reviews2[i] && i != 0 && i != students.Length - 1)
    {
        temp = reviews2[i];
        reviews2[i] = reviews2[i + 2];
        reviews2[i + 2] = temp;
    }
    //check reviewer 3
    if (students[i] == reviews3[i] && i == 0)
    {
        temp = reviews3[i];
        reviews3[i] = reviews3[i + 1];
        reviews3[i + 1] = temp;
    }
    else if (students[i] == reviews3[i] && i == students.Length - 1)
    {
        temp = reviews3[i];
        reviews3[i] = reviews3[i - 1];
        reviews3[i - 1] = temp;
    }
    else if (students[i] == reviews3[i] && i != 0 && i != students.Length - 1)
    {
        temp = reviews3[i];
        reviews3[i] = reviews3[i + 1];
        reviews3[i + 1] = temp;
    }

```

```

    }
}
try
{
    DateTime today = new DateTime();
    today = DateTime.Now;
    conn.Open();

    for (int i = 0; i < students.Length; i++)
    {
        String insertuser = "insert into Reviewers (StudentId, Reviewer1, Reviewer2,
Reviewer3, CreationDate, AssignmentId) values (@std, @rev1, @rev2, @rev3, @cdate,
@assid)";
        SqlCommand com = new SqlCommand(insertuser, conn);
        com.Parameters.AddWithValue("@std", students[i]);
        com.Parameters.AddWithValue("@rev1", reviews1[i]);
        com.Parameters.AddWithValue("@rev2", reviews2[i]);
        com.Parameters.AddWithValue("@rev3", reviews3[i]);
        com.Parameters.AddWithValue("@cdat", today);
        com.Parameters.AddWithValue("@assid", AssId);
        com.ExecuteNonQuery();
    }
    conn.Close();
    Page.ClientScript.RegisterStartupScript(this.GetType(), "Scripts", "<script>
alert('Reviewer successfully assigned based on last assignment') </script>");
}
catch (Exception ex)
{
    Response.Write("Error:" + ex.ToString());
}
}
protected void Answer(object sender, EventArgs e)
{
    DataTable dt = new DataTable();
    SqlDataAdapter adp = new SqlDataAdapter();
    int id = int.Parse((sender as Button).CommandArgument);
    {
        SqlCommand cmd = new SqlCommand("select * from Assignments where
AssignmentId=@Id ", conn);
        cmd.Parameters.AddWithValue("@Id", id);
        adp.SelectCommand = cmd;
        adp.Fill(dt);
        if (dt.Rows.Count > 0)
        {
            SubmitAnswer.Visible = true;
            lblAssName.Text = dt.Rows[0]["AssignmentName"].ToString();
            lblAssDesc.Text = dt.Rows[0]["Description"].ToString();
            lblAId.Text = dt.Rows[0]["AssignmentId"].ToString();
        }
    }
}
catch (Exception ex)
{

```

```

        ScriptManager.RegisterStartupScript(this, this.GetType(), "Message", "alert('Error
occured : " + ex.Message.ToString() + "');", true);
    }
    finally
    {
        dt.Clear();
        dt.Dispose();
        adp.Dispose();
    }
}
protected void GetCriteria()
{
    int i = ViewState["cnt"] != null ? (int)ViewState["cnt"] : 0;
    int AssId = Convert.ToInt32(dropAssignment.SelectedValue.ToString());
    SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["PeerReviewConnectionString"].C
onnectionString);
    conn.Open();
    da = new SqlDataAdapter("select * from Criteria where AssignmentId=" + AssId +
"", conn);
    SqlCommandBuilder builder = new SqlCommandBuilder(da);
    ds = new DataSet();
    da.Fill(ds, "Criteria");
    if (ds != null)
    {
        if (ds.Tables[0].Rows.Count > 0)
        {
            lblcriteria.Text = ds.Tables[0].Rows[i]["Description"].ToString();
            lblCritId.Text = ds.Tables[0].Rows[i]["CriteriaId"].ToString();
            lblAssId.Text = ds.Tables[0].Rows[i]["AssignmentId"].ToString();
            weight=Convert.ToInt32(ds.Tables[0].Rows[i]["Weight"].ToString());
            //lbl.Text = (i + 1).ToString();
            lblperf.Visible = false;
            lblperf.Text = weight.ToString();
            lblcnum.Text = ((i + 1) + " of " + ds.Tables[0].Rows.Count).ToString()+"
Criteria";
            ViewState["cnt"] = i;
            i = i + 1;
        }
    }
}

protected void btnNext_Click(object sender, EventArgs e)
{
    if (rdLow.Checked == false && rdMedium.Checked == false && rdHigh.Checked
== false)
    {
        perf = 0;
    }
    else if (rdLow.Checked == true)
    {
        perf=40;
    }
}

```

```

    }
    else if (rdMedium.Checked == true)
    {
        perf = 70;
    }
    else if (rdHigh.Checked == true)
    {
        perf = 100;
    }
    if (perf == 0 || txtcomment.Text=="")// if no performance is selected, prompt the user
    {
        ScriptManager.RegisterStartupScript(this, this.GetType(), "Message", "alert('You
must select a level and give a comment');", true);
    }
    else
    {
        SaveCriteriaAnswer();
        int i = ViewState["cnt"] != null ? (int)ViewState["cnt"] : 0;
        i = i + 1;

        int AssId = Convert.ToInt32(dropAssignment.SelectedValue.ToString());
        SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["PeerReviewConnectionString"].C
onnectionString);
        conn.Open();
        da = new SqlDataAdapter("select * from Criteria where AssignmentId=" + AssId +
"", conn);
        SqlCommandBuilder builder = new SqlCommandBuilder(da);
        ds = new DataSet();
        da.Fill(ds, "Criteria");
        if(i<ds.Tables[0].Rows.Count)
        {
            lblcriteria.Text = ds.Tables[0].Rows[i]["Description"].ToString();
            lblCritId.Text = ds.Tables[0].Rows[i]["CriteriaId"].ToString();
            lblAssId.Text = ds.Tables[0].Rows[i]["AssignmentId"].ToString();
            weight = Convert.ToInt32(ds.Tables[0].Rows[i]["Weight"].ToString());
            lblperf.Visible = false;
            lblperf.Text = weight.ToString();
            lblcnum.Text = ((i + 1) + " of " + ds.Tables[0].Rows.Count).ToString() + "
Criteria";

            ViewState["cnt"] = i;
        }
        else
        {
            lbl.Text = "Done";
            btnNext.Enabled = false;
            ScriptManager.RegisterStartupScript(this, this.GetType(), "Message",
"alert('You are done with this review. Thanks');", true);
        }
    }
}
protected void SaveCriteriaAnswer()

```

```

{
    int wei = Convert.ToInt32(lblperf.Text);
    OverPerformance = (perf * wei)/ 100 ;
    try
    {
        DateTime today = new DateTime();
        today = DateTime.Now;
        conn.Open();
        String insertuser = "insert into CriteriaAnswers (CriteriaId, AnswerId, AssignmentId,
Performance, Comment, UserId, ReviewerId) values (@crid, @ansid, @assid, @per, @com,
@uid, @rev)";
        SqlCommand com = new SqlCommand(insertuser, conn);
        com.Parameters.AddWithValue("@crid", lblCritId.Text);
        com.Parameters.AddWithValue("@ansid", lblAnsId.Text);
        com.Parameters.AddWithValue("@assid", lblAssId.Text);
        com.Parameters.AddWithValue("@per", OverPerformance);
        com.Parameters.AddWithValue("@com", txtcomment.Text);
        com.Parameters.AddWithValue("@uid", lblreviewee.Text);
        com.Parameters.AddWithValue("@rev", Session["userId"].ToString());
        com.ExecuteNonQuery();
        txtcomment.Text = "";
        rdHigh.Checked = false;
        rdLow.Checked = false;
        rdMedium.Checked = false;
        Page.ClientScript.RegisterStartupScript(this.GetType(), "Scripts", "<script>
alert('Reviewer successfully assigned') </script>");
        conn.Close();
    }
    catch (Exception ex)
    {
        Response.Write("Error:" + ex.ToString());
    }
}

```