COLLEGE OF COMPUTING AND INFORMATION SCIENCES

A programming assignment autograder for moodle

By

CS18-02

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

A Project Proposal Submitted to the School of Computing and Informatics Technology For the Study Leading to a Project Report in Partial Fulfillment of the Requirements for the Award of the Degree of Bachelor of Science in Computer Science Of Makerere University

Supervisor

Ernest Mwebaze

School of Computing and Informatics Technology, Makerere University

emwebaze@gmail.com, +256-772-121-272

October, 2017

GROUP MEMBERSHIP

SNo	Names	Registration Number	Signature
1	ABULE AUGUSTINE ARUMADRI	15/U/2633/PS	
2	AYIKO JEREMIAH SARA	15/U/186	
3	OWOMUGISHA ISAAC	15/U/12351/PS	
3	SENDIKADDIWA MARVIN	15/U/1154	

Contents

1 Introduction

1.1 Background

E-learning is a style of teaching and learning done over a network. It is independent of location and time constraints in regard to attendance. As soon as the materials (lectures notes, assignments, etc.) are uploaded, a student can access them at any time. It offers a way to design courses in a more interactive way e.g. through the use multimedia and is cost effective. Today, multiple universities all over the world use e-learning platforms to enhance understanding of course content such as Harvard, Stanford, and MIT [**?**]. With time more efficiently spent, multiple assignments can be given and automatically graded to identify and improve areas that the students did not understand fully.

Moodle is an e-learning platform designed for educators, administrators and learners with a single robust, secure and integrated system to create personalized learning environments [**?**]. It is easy to use and open source. Moodle will be used as the core system on which the e-learning platform will be designed. Makerere University already uses Moodle to run it's e-learning platform called MUELE [**?**]. We are building the system on top of moodle because of the overwhelming challenges involved in building a system from scratch (such as security considerations and user management issues).

1.2 Problem Statement

Monitoring each student's performance in Makerere university's classes is a tiresome task for lecturers. When students are given a programming course work, they expect fair and continuous feedback from their lecturers in the least time possible. However, at times students are given complex tasks to do as course work. For a lecturer to give reasonable feedback to every student's course work, he would be required to run the students code into the IDE to correctly analyze the running time, correctness and generate relevant feedback. Besides analyzing code efficiency and correctness of the code, the lecturer is also subjected to the task of checking for plagiarism among the submitted code. The current prevailing solution to improve the lecturer's productivity is grouping students. The problem with this is that only a few students contribute to the course work and the rest do not benefit. This is also not a permanent solution since only group effort is monitored at the expense of individual effort.

This project's intention is to solve the above problem by creating an on-line system that will keep track of a student's individual progress in the programming field and monitoring the student's general performance in the least time possible. With this implemented, student's submitted course work will be attended to quickly, plagiarism acts will be detected, teachers' productivity will be improved and student's passion for programming and computer science will be boosted.

1.3   Objectives

1.3.1   Main Objective

The general intention of the project is to develop a Moodle plugin that will automatically grade programming assignments uploaded to the system by students, keep track of each students performance based on scores attained from previously uploaded assignments and provide relevant feedback both to the students and the lecturer. The system should also enable lecturers to give in-class assignments for real-time assessment.

1.3.2   Specific Objectives

- To do literature review on existing E-learning systems.

- To design the proposed system.

- To implement the proposed system.

- To test the system to ensure that it meets the specified requirements.

1.4   Scope

Computer programs and source code exist in many programming languages and size (in terms of actual lines of code) and they require different computer resource to execute properly. In order to support a broad range of courses, it is important to build a general platform that can support as many languages as possible. However, the aim of this project is to build a proof of concept platform. Thus, we mainly be concentrating on the C and Java programming languages. However, in the future we want the system to be platform independent.

When designing a platform used for automatic grading purposes, accurate assessment is of utmost importance. The programming assignments must therefore be tailored with the intention of keep the auto-grading process simple and and stable while at the same time challenging and relevant to students in order to hone their programming skills at the same time pass that particular course unit. Another important aspect is user feedback. The system is designed to be a substitute of the lecturer when it comes to assessment. Therefore it must provide relevant feedback to the students regarding use of better or correct algorithms, and also to the lecturer on topics where students are finding challenges.

For purposes of testing the effectiveness of the working system, we will focus on programming course units taught in the Computer Science course only within Makerere University. Programming languages covered in this course are C and Java.

1.5   Significance

A delivered working system will be of great help to both students and lecturers who use it as part of their teaching tools. Many at times students do not get the personal attention from lecturers as they so do desire. This may be attributed to the large number of students

offering a particular course unit, each most likely with their unique challenges they face. There also exists the general perception that university students are not supposed to be spoon-fed, but rather engage in rigorous research and extensive personal reading. Much as this encourages academic independence, many students need one on one academic guidance. Our auto grading system provides this guidance.

By having the ability to suggest better algorithms and language structures/functions to use, each student can be guided on better coding techniques and how to optimize their programs. In the current setup at CIT, this is very difficult to achieve through manual means by the lecturer.

By enabling the lecturer to give in-class assignments that can be automatically graded, students are encouraged to constantly practice programming skills in order to be prepared for these assignments. This has a direct positive impact on their academic performance in that particular course unit, at the same time improving their general programming skills even though not examinable.

2 Literature Review

In this section, we provide a summary the previous work that has been done on E-learning platforms, auto-graders and online judges.

Many of the leading universities of computer science education around the world incorporate auto grading in their programming and algorithms courses. For example, in 2013, MIT researchers Singh et al [**?**] introduced a new method for automatically providing feedback for introductory programming problems. According to [**?**], due to fair and timely feedback results from online judge websites, online practice outperforms traditional programming practice.

2.1 Learning Management Systems

A learning management system (LMS) is a software application for the administration, documentation, tracking, reporting and delivery of educational courses or training programs.[**?**] They help the instructor deliver material to the students, administer tests and other assignments, track student progress, and manage record-keeping. LMSs are focused on online learning delivery but support a range of uses, acting as a platform for fully online courses, as well as several hybrid forms, such as autograders. In the U.S. higher education market as of fall 2016, the top three LMSs by number of installations were Blackboard (32.3%), Moodle (18.6%) and Canvas (23.5%)[**?**]. The same three systems lead in terms of number of students enrolled, but in a different order: Blackboard (41.8%), Canvas (32.5%), Moodle (14.6%).

2.1.1 Moodle

Moodle is an e-learning platform designed for educators, administrators and learners with a single robust, secure and integrated system to create personalized learning environments. [**?**] The moodle website [**?**] states that "Moodle is a free and open-source software learning management system written in PHP and distributed under the GNU General Public License. Developed on pedagogical principles, Moodle is used for blended learning, distance education, flipped classroom and other e-learning projects in schools, universities, workplaces and other sectors." The Moodle platform is composed of several components that enable its dynamism and thus can be adopted for the project. It is one of the most dominant e-learning platform used worldwide.

2.1.2 Massive Open Online Courses

A massive open online course (MOOC) is an online course aimed at unlimited participation and open access via the web[**?**]. In addition to traditional course materials such as filmed lectures, readings, and problem sets, many MOOCs provide interactive user forums to support community interactions among students, professors, and teaching assistants. The MOOC landscape has grown to include 9,400 courses, more than 500 MOOC-based credentials, and more than a dozen graduate degrees[**?**]. According to data gathered by Class Central, around 20 million new learners signed up for their first MOOC in 2017. Thats fewer than the 23

million new learners who registered for a MOOC in 2016. To date, over 800 universities around the world have launched at least one MOOC including reknown universities such as Havard[?], MIT[?], Stanford[?].The total number of MOOC learners is now 78 million. Here is a list of the top five MOOC providers by registered users[?]:

- Coursera 30 million users.

- edX 14 million users.

- XuetangX 9.3 million users.

- FutureLearn 7.1 million users.

- Udacity 5 million users.

2.2 Methods of Delivery

2.2.1 Asynchronous

Asynchronous learning environments are described as online spaces where work is supported through the use of digital platforms in such a way that participants are not required to be online at the same time.[?] Threaded discussions, e-mail, and telephone calls are options of asynchronous delivery. This gives meaning to the anytime-anywhere appeal of online learning.

2.2.2 Synchronous

Synchronous learning environments most closely resemble face-to-face learning. Synchronous learning takes place through digital platforms where the learners are utilizing the online media at the same time. When compared to asynchronous learning, synchronous online environments provide a greater sense of feeling supported, as the exchange of text or voice is immediate and feels more like a conversation. If platforms such as web conferencing or video chat are used, learners are able to hear the tone of voice used by others which may allow for greater understanding of content

2.2.3 Flipped Classroom

Flipped classroom is an instructional strategy and a type of blended learning that reverses the traditional learning environment by delivering instructional content, often online, outside of the classroom. It moves activities, including those that may have traditionally been considered homework, into the classroom. In a flipped classroom, students watch online lectures, collaborate in online discussions, or carry out research at home and engage in concepts in the classroom with the guidance of a mentor. The flipped classroom intentionally shifts instruction to a learner-centered model in which class time explores topics in greater depth and creates meaningful learning opportunities such as in-class assignments that can easily be graded automatically and feedback given on spot.

2.3 Related work

2.3.1 MUELE

MUELE (Makerere University E-learning Environment) [?] is the dominant e-learning platform used at Makerere University. It is used for publishing notes and assignments for students. It also has functionality for giving on-line quizzes (mainly objective type and single-response questions). MUELE is based on the moodle online learning platform.

2.3.2 Building plugins for Moodle

Plugins are a flexible tool set, allowing Moodle users to extend the features of the site" [?]. Plugins for moodle can be thought of as modules or widgets with specific functionality. Currently moodle is consists of thousands of plugins developed to suite different teaching curriculum. Some of the existing plugins relevant to our project are: URKUND plagiarism plugin [?] used for checking plagiarism in submitted files and Reengagement plugin [?] for sending timely feedbacks to remind students of uncompleted course activities. Since some of these plugins are open source, they can be used as sub modules in our project.

Moodle has a well-structured documentation to help developers write plugins from start to publishing them on the website for usage. The prerequisites for plugin development are: -

- Comfort using (Hypertext preprocessor file) PHP.

- Ability to install a database and a web server on a local machine.

Resources for programmers with no skills of prerequisites are provided in the documentation. These include PHP tutorials for novice programmers and database installation video tutorials links to YouTube. If you download Moodle source code or clone it from git, you will see a bunch of files and folders. This code consists of Moodle core (that consists of the Very core and Core components), third party libraries and plugins. Moodle has 24 different categories of plugins[?] and uses git for development. If one wishes to develop a plugin and finds out it has already been developed, then they may opt to contribute to its code. For one to develop a new kind of plugin, a skeleton of a plugin is provided (template) with basic PHP code that displays a simple page. This only contains the page API which all plugins that display something on the webpage include. The documentation continues to explain the cloned repository file structure, callbacks, Core APIs, adding basic contents to your page, to publishing your plugin into Moodles Navigation. A further documented tutorial to manipulate Databases in Moodle is provided for advanced plugins development.

2.3.3 Plugins for Moodle

Moodle has thousands of plugins each falling in different categories/types. Some of these are :-

- Anti-viruses: which check for viruses in uploaded submitted files. Antivirus scanner plugins provide functionality for virus scanning user uploaded files using third-party virus scanning tools in Moodle. For example: ClamAV.

- Quiz reports: For displaying and analyzing the results of quizzes, or just plug miscellaneous behavior into the quiz module.

- Question types: Different types of question (e.g. multiple-choice, drag-and-drop) that can be used in quizzes and other activities.

- Question behaviors: Control how student interact with questions during an attempt.

- Plagiarism plugins: Define external services to process submitted files and content. The files handled include reports and programming work.

As far as programming assignment plugins are concerned, the only plugin developed is Programming Code Plagiarism. This plugin integrates two well-known source code plagiarism detectors MOSS and JPlag into Moodle to compare the similarities of programming assignments of students within the same course. By integrating these detectors seamlessly into Moodle, it eliminates the overhead of manually preparing the assignments to feed into these detectors, present the reports with an interactive interface and, more importantly, serve as an educational tool to provide feedback to students and raise their awareness of plagiarism. The plugin has a comparison view that looks for similarities in work done by one student to the rest of the students. By ticking the checkbox "Show similarity between someone with other students", lecturers can also see the similar parts of that students with everyone in the class (not just pair to pair).

2.3.4 Auto-graders/Assessment plugins for Moodle

On Moodle, a lecturer/instructor can add an assignment activity for students. This involves describing the assignment nature where a series of steps are attended to in a chronological order viz. Assignment name, description, Due date, require students to click submit button, etc. As far as our project is concerned, the most interesting part on the assignment settings is Grading. There are three options to choose a grading method. Viz. simple direct grading (entering a grade or scale item), Marking guide and Rubric.

Marking guide: A marking guide is an advanced grading method where a teacher enters a comment per criterion and a mark up to a maximum. This involves creating a marking guide that is to be followed by students and markers in order to know what is required and the maximum mark per criterion is provided too. This process involves creating a set of criterions and to avoid duplicates, criterions can be added to frequently used criterions section. This is a manual marking guide that the teacher will follow after students submit their solutions.

Rubric: Rubrics are advanced grading forms used for criteria-based assessment. The rubric consists of a set of criteria. For each criterion, several descriptive levels are provided. A numerical grade is assigned to each of these levels. The rater chooses which level answers/describes the given criterion best. The raw rubric score is calculated as a sum of all criteria grades. The final grade is calculated by comparing the actual score with the worst/ best possible score that could be received.

2.3.5 Online judges and autograders

The most popular application of online judges is in programming contests such as TopCoder, ACM/ICPC and Google Code Jam. The most prominent of these is the ACM/ICPC [**?**]. The ACM/ICPC consists of algorithmic and programming problems for contestants to solve. The contestants' submissions are evaluated based on the output of their programs. A typical problem in this contest consists of the following parts:

- Problem description: This is a description of the problem to be solved.

- Input description: This gives the specification of the input file from which the users solution programs read data

- Output description: This specifies the format in which the program should produce results.

- Sample input and sample output: These provide examples to clarify the input and output specifications.

For each problem in ACM/ICPC, there are multiple sets of data to be processed. This is an important feature that serves two major purposes [**?**]: first, they are used to test that the program can work in all possible situations (including corner cases). Second, they are used to calculate the running time of a program (a good solution should be able to work on both small datasets and larger datasets). The use of multiple datasets therefore ensures that a students submission is evaluated comprehensively.

Test datasets of problems in the ACM/ICPC are either generated by hand for small-sized datasets or by programs written by jury members that generate datasets according to some predefined patterns or at random [**?**].

In [**?**], Dr. Antti Laaksonen describes how a competitive programming approach using an online judge to teach algorithm concepts has helped improve the problem solving skills of the students at the University of Helsinki. In the course, the TMC system [**?**] is used. TMC allows students to create their Java solutions in the NetBeans IDE and evaluates the solutions using JUnit tests. The tests are written by the course staff. In this paper, he describes how this model of teaching helps students understand the importance of developing efficient algorithms. He also points out some limitations of using online judges. He gives an example of the following problem from [**?**]:

Describe an $O(n)$-time algorithm that, given a set S of distinct numbers and a positive integer $k \leq n$, determines the k numbers in S that are closest to the median of S.

He points out the difficulty of automatically determining whether a students submission of the above problem is really $O(n)$, since the same problem can be solved in $O(nlogn)$ time. This therefore means that auto-graders are not perfect, however they seem to be better than traditional means of evaluation as the following section shows.

2.4 Benefits of auto-grading and practice oriented teaching of programming

Researchers Wang GP et al. conducted two sets of experiments [**?**] in order to evaluate the effectiveness and validity of using their online judge system called OJPOT as compared to

traditional teaching methods. They set out to answer the following questions [**?**]:

- Does OJPOT work effectively to enhance students' practical abilities compared to the traditional teaching idea?

- In which aspects can OJPOT improve the student's practical abilities?

- Output description: This specifies the format in which the program should produce results.

In the first set of experiments, two classes were chosen: one class defined as the control class (CC) and another defined as the experimental class (EC). In the CC, the traditional teaching idea was applied; while in the EC, the OJPOT online judge was used [**?**]. Before the experiment, both classes were given a pre-test to evaluate their elementary knowledge in C programming. Then during the semester, both classes were taught by the same teacher with the same timetable. The following data were collected during the experiment: pre-test scores, online practice and statistical data, post-test scores and course project scores. Below is a summary of the results of their experiment ([**?**]):

- Both classes had similar programming foundation before the experiment

- At the end of the semester, there were significant differences between the CC class and the EC class: the EC class has overall better results in the final tests and project.

They conclude their results by remarking that the EC class performed better because the online judge system made the students more enthusiastic about the material they were studying. It was also observed that the general programming abilities of the EC class were improved much better than those of the control class.

2.5 Conclusion

In computing, time is an important aspect. This can be taken care of if we avoid creating already existing technology reinventing the wheel. If we are to create our own project from scratch it will be tedious. We will be required to design a whole complete site with a perfect interface and backend functionality for the project to be viable. With the presence of existing open source projects like the one cited above i.e. Moodle, we can collaborate with them and contribute to their source code. In so doing, the advantages encountered will be:-

- More time to concentrate on the projects main functionality and objectives.

- After development, the project will not be rendered trivial. Moodle being a popular platform worldwide, our efforts of marketing for a system would be exponentially reduced to zero.

- The main institute we are targeting to use our projects deliverable (plugin) is Makerere University. As stated earlier, this institute has its own E-Learning Environment (MUELE) which also runs on the Moodle framework. Thus, implying that it can perfectly run any plugin run by the Moodle website.

3 Methodology

3.1 Data collection Methods and Tools

Expected requirements of the plugin will come from sources like reading articles about online contests, searching and using online similar platforms (both educational and non-educational based), designing and distributing interviews to Makerere Lecturers that teach course units in the context of this problem. All this will help us agree in a common ground of designing a system that is in line with their expectations.

3.1.1 Literature Review

This will involve researching on the internet for relevant information and data, reading literature about existing E-learning platforms with or without auto grader functionality. The information obtained from such documentation will in part be necessary for the identification and analysis of the requirements needed to accomplish the implementation of the system, and for coming up with strategies to take advantage of the weaknesses of the existing systems.

3.1.2 Interveiwing

Interviews will be made to acquire knowledge through seeking for advice from experienced instructors and experts as well as people with knowledge about Blended learning, flipped classrooms and how they find the existing systems that support them.

3.1.3 Questionnaires

A few programming course instructors at Makerere University will take part of the study where they will be handled forms that will involve questions on how best the system to be designed and their expectations from it.

3.2 Designing the system

Below is a use case diagram showing the interaction between the users and the main features of the system.
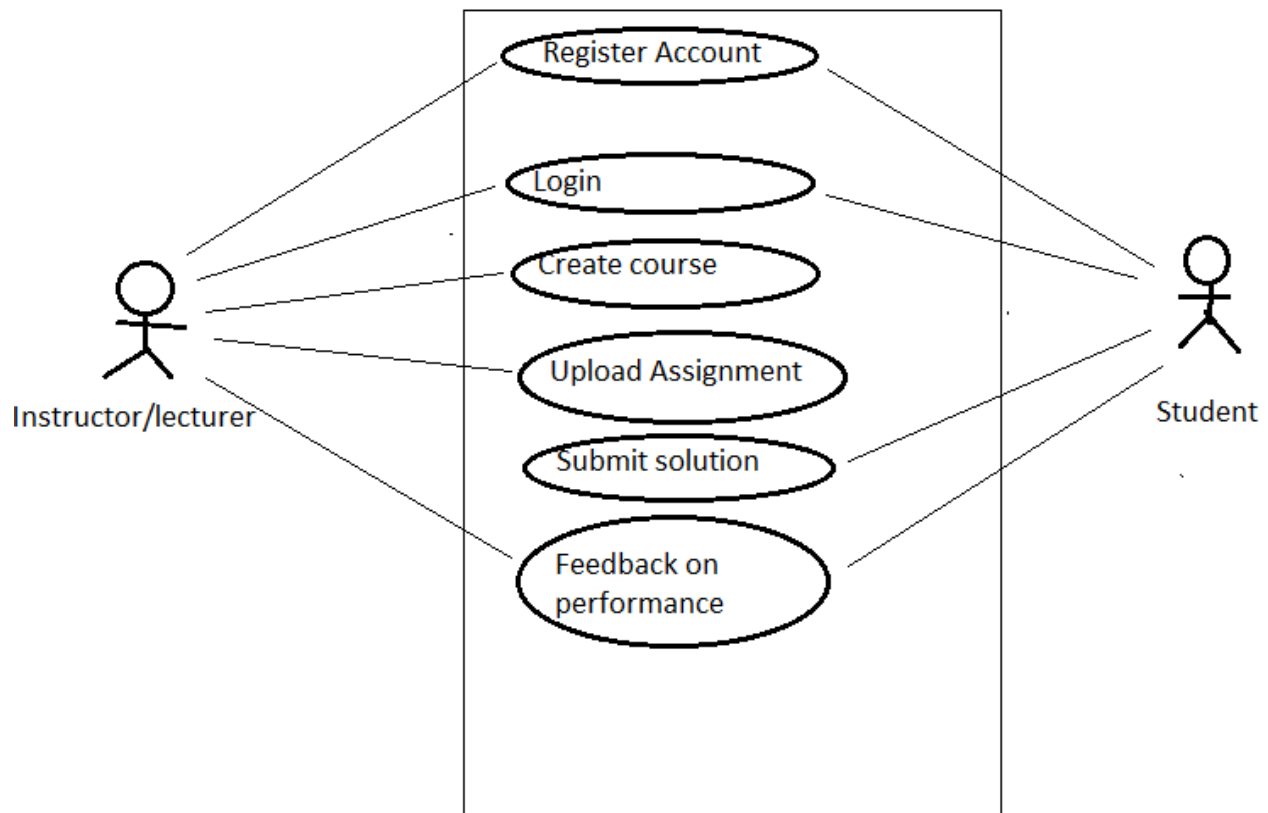
3.2.1 UML Diagrams

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system. We will use UML to design the system. Below are the UML diagrams that are going to be used in designing the system.

- Class diagram: A type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

- Use case diagram: a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

- Sequence Diagram: An interaction diagram that shows how objects operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence.

3.3 Implementing the Proposed System

3.3.1 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as[?]:

- A flexible Gradle-based build system

- A fast and feature-rich emulator

- A unified environment where you can develop for all Android devices

- Instant Run to push changes to your running app without building a new APK

- Code templates and GitHub integration to help you build common app features and import sample code

- Extensive testing tools and frameworks

- Lint tools to catch performance, usability, version compatibility, and other problems

3.3.2 NetBeans

NetBeans IDE 8.2 provides out-of-the-box code analyzers and editors for working with the latest Java 8 technologies–Java SE 8, Java SE Embedded 8, and Java ME Embedded 8. The IDE also has a range of new tools for HTML5/JavaScript, in particular for Node.js, KnockoutJS, and AngularJS; enhancements that further improve its support for Maven and Java EE with PrimeFaces; and improvements to PHP and C/C++ support.[?]

We will use netbeans to write our PHP code.

3.3.3 Wamp Server

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your databases.

We will use wamp server to develop the database that will be used in the system. The database will be written using MySql.

3.4 Other Software and Applications for Development

3.4.1 Github

GitHub is a web-based hosting service for version control using git. It is mostly used for computer code. It offers all of the distributed version control and source code management

(SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project[**?**].

We will use github to store our source code and documents as well as keep track of the changes made on them.

3.4.2 Latex

LaTeX is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. LaTeX is the de facto standard for the communication and publication of scientific documents[**?**].

We will use latex to write all the documentation of the system.

3.5 Testing the system

We plan on testing the system in a real classroom environment. Once the system is functional, we shall request a lecturer who teaches a programming related course unit to try out the system in his class. In the testing phase, we shall seek to answer the following questions:

- Does the system meet the functional and non-functional requirements (which are highlighted in the software requirements specification document)?

- Can the system be used without issues for giving assignments? This is to determine the system reliability and usability. (This will be determined by monitoring logs for crashes and other issues that may arise).

- Does the system improve on how programming assignments are given in Makerere University as compared to current methods? (This shall be determined by giving surveys to the students and the lecturers.)

In addition, we shall write comprehensive unit-tests to ensure that the system has expected behavior.

4   Appendices

4.1   Appendix A: Activities Gantt chart
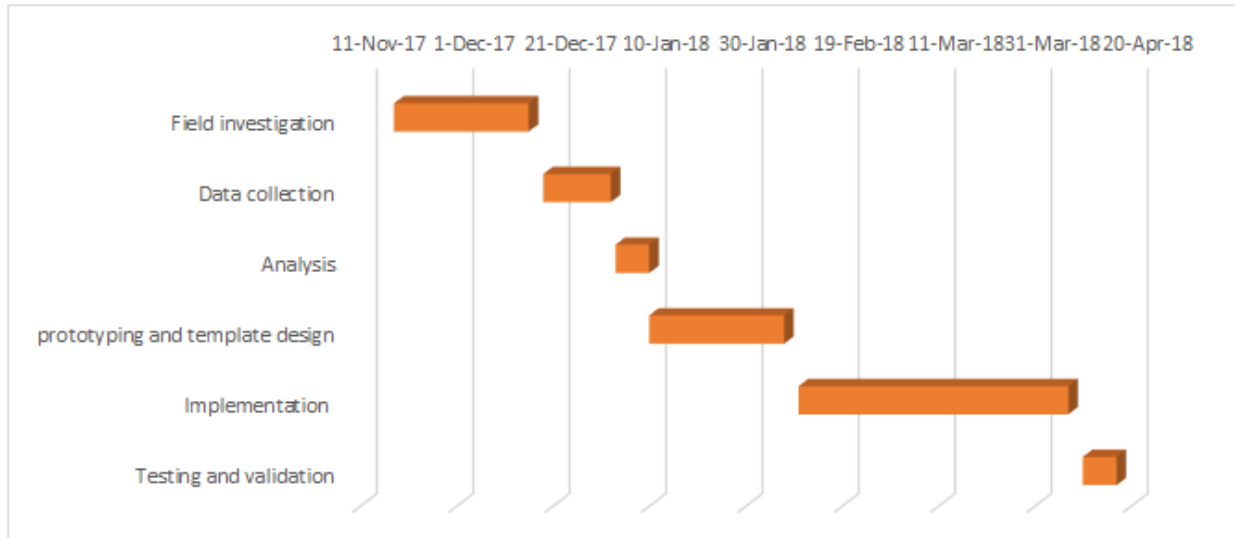
4.2   Appendix B: Financial Requirements

Requirement	Quantity	Unit price (UShs)	Total price
Generating questionnaires	50	1,000	50,000
Data(Internet)	20gb	12,000	240,000
Private GitHub repository	1	32,400(per month)	64, 800
Miscellaneous	-	100,000	100,000
Total			454,800