International Journal of Scientific Engineering and Science
Volume 2, Issue 4, pp. 83-90, 2018. ISSN (Online): 2456-7361

Development of an Android-Based Class Record for Teachers

Dave E. Marcial[1], Chuchi S. Montenegro[2], Ranel B. Ubarre[3], Melody Angelique C. Rivera[4], Marianne S. Yao[5], Fairy May G. Katada[6]
[1, 2, 3, 4, 5, 6] College of Computer Studies, Silliman University, Dumaguete City, Negros Oriental, Philippines
Email address: [1]demarcial[AT]su.edu.ph

Abstract*—Mobile technology redefines assessment and classroom management. It allows teachers to become more mobile, collaborative and innovative. This paper describes the development of a mobile class record application for higher education institutions in the Philippines. The application is called mClassRecord, a stand-alone mobile application that automates the manual process of a class information recording and management practices of teacher educators in the Philippines. It only runs on Android platform. The development employs the feature-driven approach specifically during the design and coding phases. The app allows the teacher to record students, manage class information, monitor attendance, assign groupings, manage tasks, assess students, and view a dictionary of teacher education terms. Test cases were used during system testing, while the PACMAD model was used during the usability testing. The evaluation shows that mClassRecord is highly usable. The study concludes that the development of the mobile class record application is a success. Integrating many characteristics of a class record in one app is a challenging development task. The study recommends making mClassRecord online to incorporate data-syncing feature.*

***Keywords*—** *Android; class record; ICT in Education; mobile development.*

I. INTRODUCTION

Record-keeping refers to any prearranged means to record, then appropriately share, the class standing of the students [1]. It tracks student's progress, proves student's achievement, identifies issues such as low attendance and learning difficulties, and ensures all sections of the course have been completed [2]. "Developing good habits is essential for organizing, processing, and communicating the students' understanding of the curriculum" [3].

Oberleitner and Abowd [1] believed that computer and Internet technologies would make record keeping easier, efficient, and effective. It provides pedagogical benefits. It changes the landscape in assessment and classroom management. Most of all, it redefines class and management. The conventional use of ICT in class record keeping includes the use of word processing and spreadsheet, among others. With the advent of mobile technologies, classroom management and class are now becoming more flexible and portable. Mobile applications are now a challenging issue in the education sector.

Griffith [4] stresses that "students are not the only ones bringing electronic devices to school." He means that some teachers are also adept at technology amidst the technological gap between them and the students. PBS LearningMedia [5] discloses that almost 74% of all teachers surveyed said they use teaching digital tools like tablets. Likewise, Marcial [6] revealed that a mobile class record application is perceived to be highly acceptable. In particular, the acceptability of the teacher educators towards the development of a portable classroom manager is very high, positive, and encouraging. Marcial [6] concluded that teacher educators in Central Visayas, Philippines will strongly adopt the proposed instructional tools and integrate mobile learning into their teaching and learning activities. This result motivated the proponent to develop a mobile class record application.

This paper presents the development of a mobile class record application. The mobile app is called mClassRecord. Specifically, this article discusses the analysis, design, and implementation of PLMS. "Systems analysis is a process that provides system requirements description; defines the system functional requirements, and requirement specification of the proposed system" (Bruegge & Dutoit, 2000, cited in [7]. The list of user requirements is presented in this article. On the other hand, systems design provides the data structures, architecture, interfaces, and procedural detail of the software component of the research/project (Pressman, 2001, cited in [7]. In this paper, use case, entity-relationship diagram, and screenshots are presented. Further, systems implementation is the process that includes planning the integration, installation of the database, testing the system, training end-users and other production activities (Bruegge & Dutoit, 2000, Whitten, Bentley, & Dittman, 2004, cited in [7]). In this article, a summary of test case results, training evaluation, and usability test are briefly explained.

II. METHODS

A. Development Framework

The study utilized a customized software development model where traditional and non-traditional development models are combined. It consists of five major phases; these are user requirements analysis, customization and coding, system testing, training and quality testing, and delivery and dissemination (Fig. 1).

User Requirements Analysis. The quantitative data on the perceived usefulness and ease-of-use of mClassRecord [6] serve as input to the user needs analysis. Notably, mClassRecord is seen to be useful and perceived to be slightly easy to use. Specifically, the respondents agree that the tool would enable them to accomplish tasks more quickly, improve their performance, productivity, and effectiveness, and believe that the tool is useful in their job. On the other hand, the

respondents somewhat agree that the tool would be natural to them, clear and understandable, flexible and easy to use.

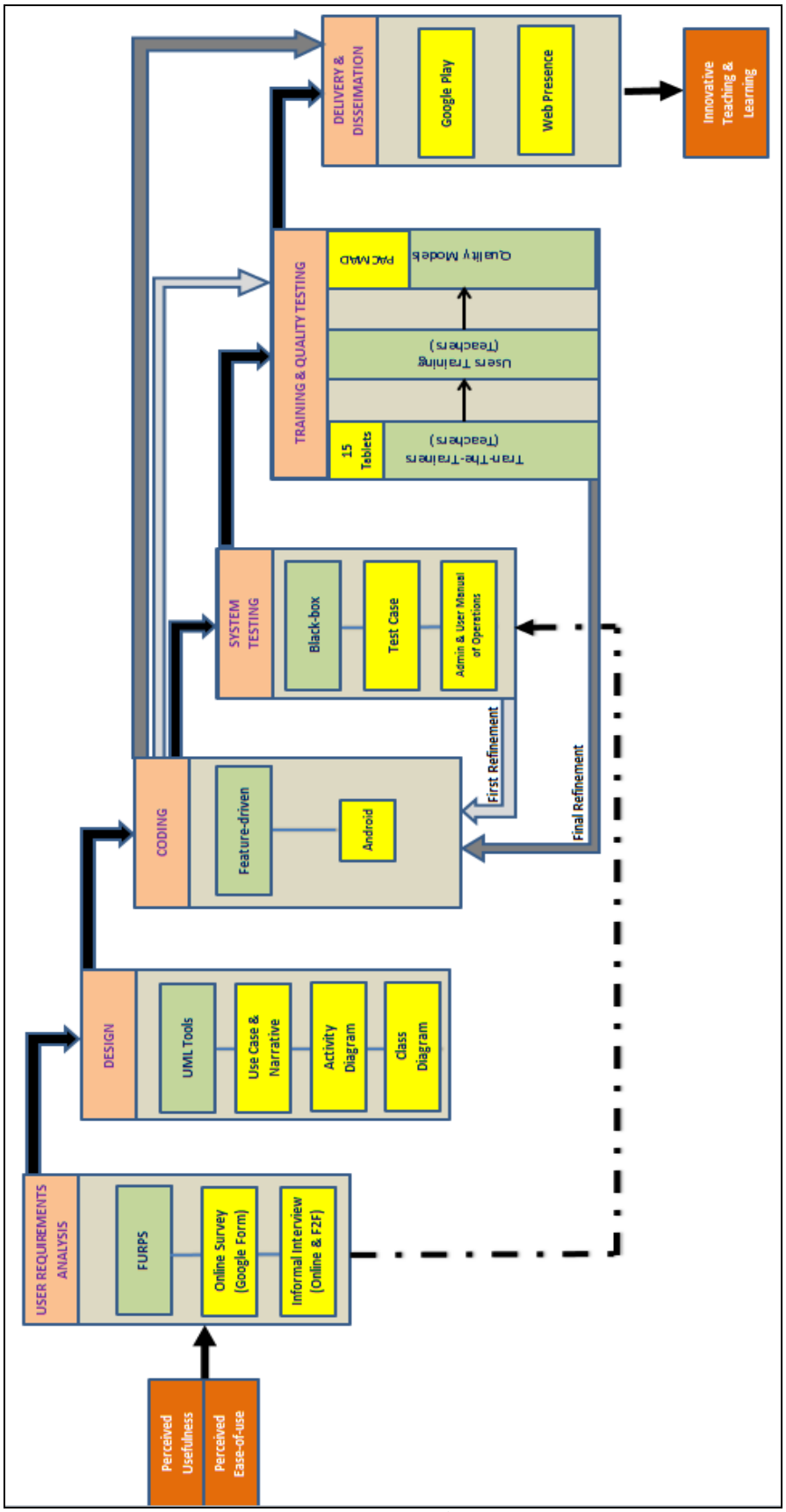

Fig. 1. The development framework.

During the elicitation of functional and non-functional requirements, the FURPS (functionality, usability, reliability, performance, supportability) model by Robert Grady at Hewlett-Packard, cited in [8] was thoroughly followed. An informal interview, both online and in person, was conducted. In a separate article, Marcial [9] describes the actual elicitation of user requirements. The elicitation was carried out using a survey questionnaire. Specifically, the respondents were asked about the need level of the proposed features of mClassRecord. A total of 122 teacher educators from Bohol, Cebu, Negros Oriental, and Siquijor were the respondents of the study.

Customization and Coding. This study utilizes the feature-driven approach during the customization and coding phases. Feature-Driven Development (FDD) "is a client-centric, architecture-centric, and pragmatic software process" [10]. FDD "is an agile, highly adaptive software development process that is highly and short iterative" [11]. There are five primary activities in FDD, adopted from [11], which were performed iteratively. These are (a) develop an overall model, (b) build a features list, (c) plan by feature, (d) design by feature, and (e) build by feature. Several meetings were also conducted in every activity.

System Testing. A black box testing, sometimes called as behavioral testing, was the technique used. Techopedia.com defines black box testing as "a software testing technique that focuses on the analysis of software functionality, versus internal system mechanisms". A team of testers was created to test mClassRecord. The testers used 49 test cases as a guide using the format by [12]. Findings from the testing were given to the programmer for action and revisions of the specific features. Then, a user's manual was developed.

Training and Quality Testing. Training is one of the critical activities during the implementation phase of software development cycle [13]. It is a factor for the successful implementation of any software [14]. There are two phases of training conducted, the train-the-trainers and end-user training. The train-the-trainers training was a 3-day activity with the aim of identifying a pool of trainers. Seventeen teachers were selected as trainers-ambassadors of mClassRecord. On the other hand, the end-user training is aimed to demonstrate, practice, and do hands-on activities with the use and integration of the newly developed digital teaching tools into the classroom. The participants of the said training must be willing to use and integrated the tools into their teaching instruction. A total of 74 end-users were trained. After the end of the end-user training, a survey on the usability evaluation was conducted. The survey questionnaire was based on the PACMAD (People at the Centre of Mobile Application Development) usability model by [15].

Delivery and Dissemination. mClassRecord is distributed for free not only in the Central Visayas but also to its neighboring regions. It can be downloaded from Google Play Store. A user manual can also be downloaded from its official website. The app is applicable for all levels of education and all types of learning.

B. Development Tools

The integrated development environment (IDE) used is the Android Studio. The emulator, Bluestacks App Player, was also used to enable the application run on Windows PCs, especially during testing processes. A Samsung Galaxy 4, an Android v2.2.2 OS with an 8GB internal memory, was the device considered especially during the interface design. The Android platform was chosen because many of the intended users have Android-based tablet computer [16].

III. Analysis, Design and Implementation

A. The User Requirements

All the proposed features are rated highly needed. The result implies that all specifications are imperative to the respondents, and all functionalities are necessary for the proposed mobile class record application. Thus, it is imperative for the project team to incorporate all the processes and information as required by the targeted users. The result signifies that the respondents extremely wanted a tool that can monitor the outcomes and performance of their learners. The identified functionalities are necessary for the overall quality of the proposed mobile class record application. There is a high degree of satisfaction of the stated needs in the proposed mobile class record. The teachers believed that the identified functionalities could improve the quality of their teaching job. They also believed that if all functionalities are to be integrated, the mobile app can perform class recording tasks efficiently and effectively. See Table I for the list of the functionalities.

TABLE I. The user requirements.

1. mClassRecord shall manage classroom information
   a. Teachers can add & update many classes in a semester
   b. Teachers can add & update class information
   c. Teachers can select & back-up/transfer externally all information in the class
2. mClassRecord shall monitor attendance
   a. Teachers can store daily attendance, including excused absences
   b. Teachers can add & update comment/s in every input of absence
   c. Teachers can view the summarized attendance with remarks/warning
   d. Teachers can visually represent by way of highlighting the records of students who are marked NA (Never Attended), WD (Withdraw), NC (No Credit), INE (Incomplete Exam), etc.
3. mClassRecord shall calculate test scores
   a. Teachers can add & update written exam information
   b. Teachers can add & update performance-based test information
   c. Teachers can add & update criteria for performance-based test
   d. Teachers can add & update obtained score for both written and performance-based
   e. Teachers can view the percentage rating per exam/performance of the student
4. mClassRecord shall contain a dictionary of Teacher Education Terms
   a. Teachers can search Teacher Education word, then view its meaning
5. mClassRecord shall offer task management
   a. Teachers can add a task, date, time, description, venue
   b. Teachers can be reminded automatically about the incoming task
   c. Teachers can view the summary of activities of the class
   d. Teachers can make special notations and remarks on tasks when the task is finished or completed.
6. mClassRecord shall store grade and let the teacher store comments along with it
   a. Teachers can add & update grading criteria
   b. Teachers can view calculated grades based on the criteria
   c. Teachers can overwrite grades
   d. Teachers can add a remark on every grade
   e. Teachers can visually represent by way of highlighting the records of students with failing marks
   f. Teachers can view the median grade or the quartile grades of the class.
7. mClassRecord shall contain an address book
   a. Teachers can add & update student information
8. mClassRecord shall manage groups
   a. Teachers can create/manage/modify groups in a class
   b. Teachers can manage a group or groups that comprise more than one class.
9. mClassRecord shall make a suggestion for a grade
   a. Teachers can view suggestions for grades based on previous performance
   b. Teachers can view suggestions about the percentage/rating needed to pass the subject
10. mClassRecord shall handle uploading photo for each student
   a. Teachers can add & update photo for each student

International Journal of Scientific Engineering and Science
Volume 2, Issue 4, pp. 83-90, 2018. ISSN (Online): 2456-7361

B. The Use Case

Fig. 2 is the use case diagram of the mobile class record application. It represents the interaction of teacher-user with mClassRecord. It also shows the relationship between the teacher-user and the different use cases in which the teacher is involved. From the ten functionalities required by the user, it is translated into six comprehensive modules. As shown, a teacher-user can add students to the address book, register student in the class, manage attendance, assign tasks and activities, manage test scores, and view teacher education dictionary. Note that the development team decided to remove photo uploading in the app with the consideration of the minimal hardware storage. Regarding the requirement for grade suggestion, the development team also decided to include this in the future revision of the app. It was found out that incorporating a grade suggestion – based on previous performance and suggestions for percentage/rating needed to pass the subject – entails intelligent computing development similar to an expert system or decision-support system.

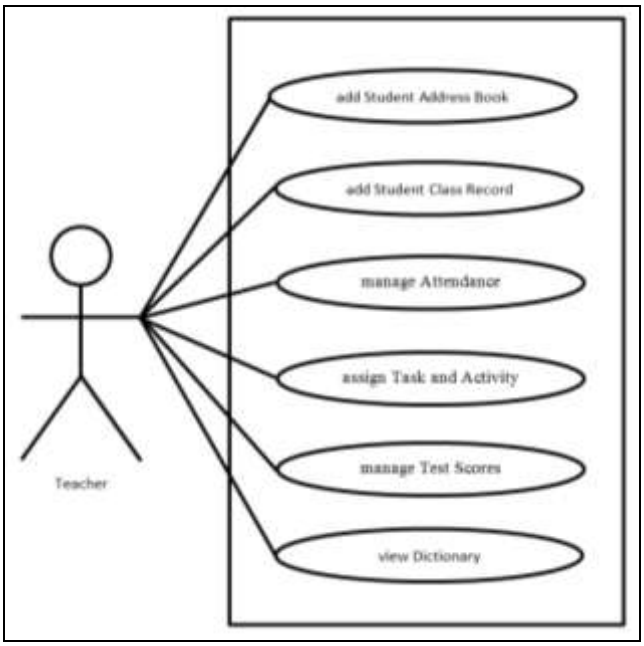

Fig. 2. mClassRecord use case diagram.

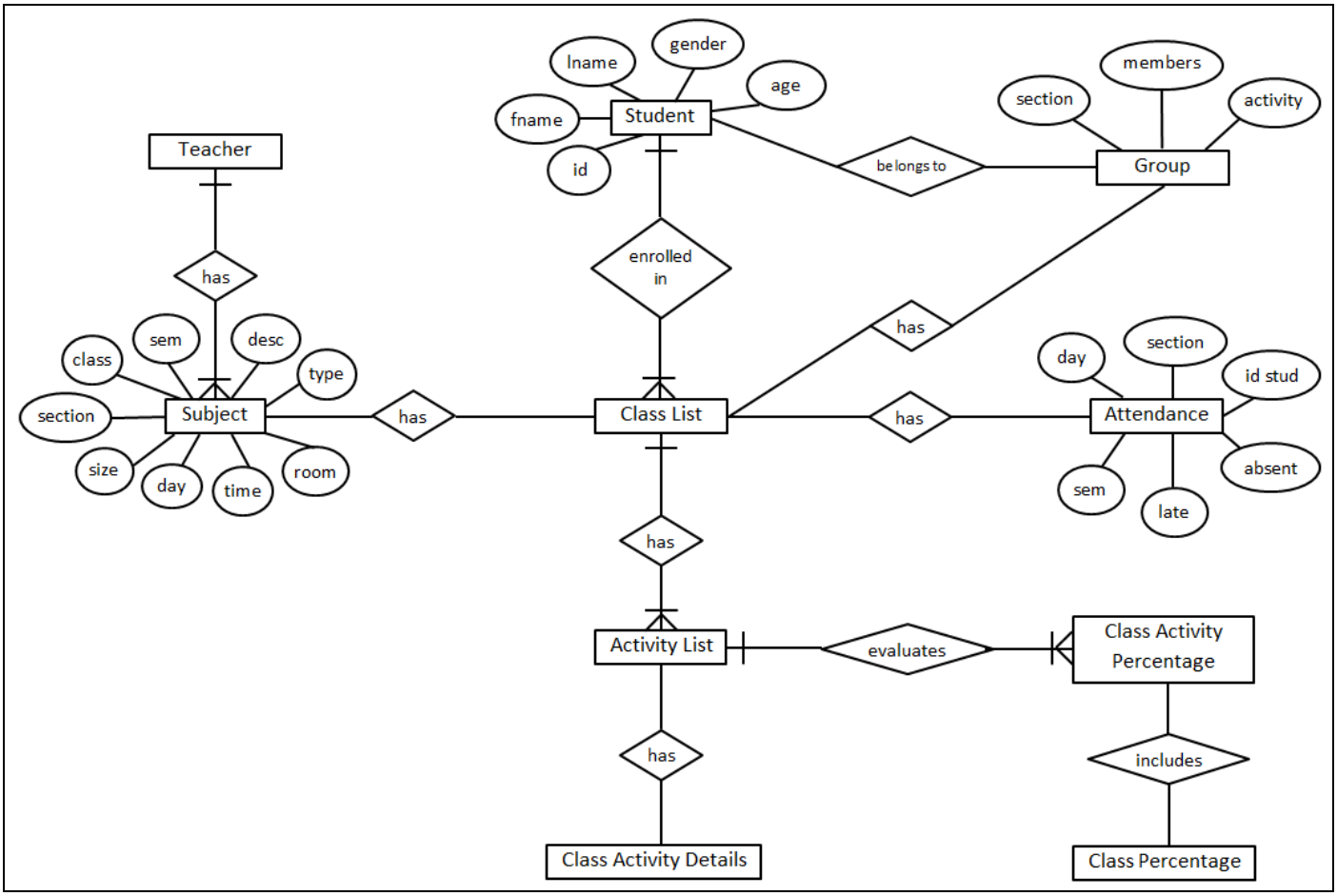

Fig. 3. mClassRecord ERD

C. The Entity Relationship Diagram

Fig. 3 is the entity-relationship diagram (ERD) that shows the entities, attributes, and relationships between the different tables in the database. It also presents the processes involved in managing the application. As shown in the diagram, there are ten entities – objects that are associated with relevant data. These are Teacher, Student, Group, Subject, Class List, Attendance, Activity List, Class Activity Percentage, Class

Activity Details, and Class Percentage. Each of these entities has, at least, one relationship with another entity. Teacher and Subject entities have a one-to-many relationship. This relationship means that a teacher-user can enrol many subjects or courses in mClassRecord. Subject information includes description, room, time, day, size, semester, class, section, and type. Class List and Subject have a one-to-one relationship. This relationship means that each class has a class list which consists of many students. Student entity and Class List have a one-to-many relationship. This relationship means that a student can be enrolled in many classes. Information regarding the students includes last name, first name, sex, age, and id number. Each student belongs to a group that contains information about section, members, and activity, showing a one-to-one relationship between the two entities. This group is part of the Class List with a one-to-one relationship. The class list also includes attendance with a one-to-one relationship having information like a day, section, student id, among others. The Class List is having a one-to-many relationship also with activity list that evaluates percentages for class. Further, Activity List has a one-to-one relationship with Class Activity Details. On the other hand, there is also a one-to-one relationship between class activity percentage and class percentage.

D. The Modules and Screenshots

The main interface of the mClassRecord (Fig. 4) comprises eight (8) modules. These are Student Records, Class Records, Attendance, Groups, Tasks, Scores, and Dictionary. See Table II for the complete description of each module.

Fig. 4. mClassRecord main interface.

The dictionary module provides a way for teachers to search and view the meaning of education-related terms (Fig. 5). The student record interface (Fig. 6) allows the teacher to add, edit, and delete in the application database student records. The search interface also provides a way for the teacher to search for students.

The class record module (Fig. 7) allows the teacher to create a class and add a student list for a particular class. It also provides an email feature where the teacher can use to send thru his/her email account all class-related information in comma-separated values (CSV) format.

TABLE II. Icons, modules, and its description of mClassRecord.

Icon	Module	Functions
	Student Records	allows the teacher to create, edit, and delete a student record
	Class Records	allows the teacher to create a class record, send and view class information, and delete a class
	Attendance	allows the teacher to check attendance, view daily and summary of attendance
	Groups	allows the teacher to add and delete group activity and student groups
	Tasks	allows the teacher to add, edit, and delete tasks and events
	Scores	allows the teacher to add, edit, and delete information about the assessments per term basis
	Dictionary	provides the teacher an interface where one can view educational terms and its meaning

Fig. 5. Dictionary interface.

Fig. 6. Student record interface.

Fig. 7. Class record interface.

The attendance module (Fig. 8) provides a way for teachers to monitor student attendance per class per day. It works by first selecting what class to check for attendance. The application will then display an attendance list comprising

of visually represented student lists sorted in an alphabetical manner. Checking of attendance can be done by just tapping on the student visual representation. The first tap means a student is late; the second tap means the student is absent. The date and time of late and absences will be derived from the device date and time settings. A feature to display a summary of attendance record is also provided in the application.

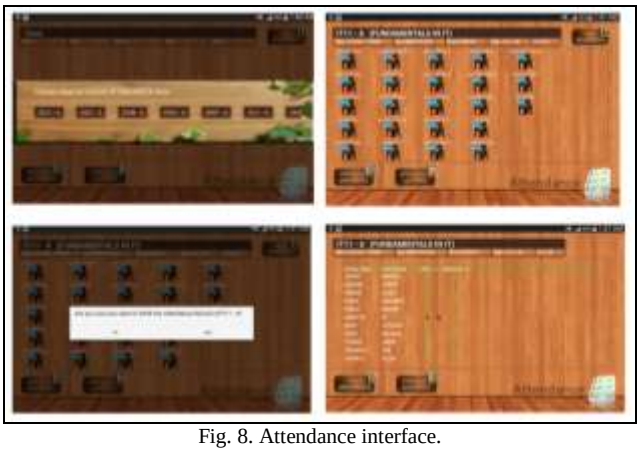

Fig. 8. Attendance interface.

The groups module (Fig. 9) allows the teacher to manage groupings per class per activity. A group can also be formed from students of different class schedules. This feature is intended to help teachers monitor groupings and at the same keep track members. The Task module (Fig. 10) functions just like a reminder or notifications. It records schedules and task and notify a user of an incoming task.

Fig. 9. Group interface.

Fig. 10. Tasks interface.

The grade module (Fig. 11) allows teachers to record student performances. The recording is done by first selecting the subject or class, term, and the type of activity.

Fig. 12 is the screenshot of the download page of mClassRecord. The app can be downloaded for free from Google Play store. The download page can be browsed at the shortened link http://bit.do/mClassRecord.

Fig. 11. Grade interface.

Fig. 12. mClassRecord download page.

E. System Test Results

The app was tested on a Samsung Galaxy 4 Tablet using 49 test cases to test all its functionalities. The major modules tested were Addressbook, Attendance, Groups, Task and Activity, Test Scores, and Dictionary. The application passed all test cases. Program testing and Link Testing were done by the pilot users from different schools. During the testing, especially in the Test Scores module, the development team found out that the schools had different grading systems, terms (e.g. Prelim, Midterm, Pre-final, Final), and class work. Because of this, the original plan for the Test Scores module was scrapped, a newly expanded module was developed, and new test cases were developed. Instead of having the mClassRecord cater to only the tertiary level, it can now be used in the basic education.

After the testing, the team met with the programmer to summarize and discuss the minor glitches of mClassRecord. Each bug or errors were solved before the train-the-trainers training. The train-the-trainers' participants also gave their comments and suggestions for improvement of PLMS.

Train-The-Trainers Training Outputs. There are three major deliverables during the train-the-trainers training. These are an action plan, terms of engagement, and a list of comments and suggestions. The action plan consists of four interrelated objectives. The first goal is to assist the launching activity. Second is to integrate mClassRecord in the classroom, and the third is to cooperate with the effectiveness evaluation of the newly developed teaching tool. To promote

mClassRecord is the final objective of the action plan. Similarly, Terms of Engagement were signed. This contract was made between the project leader and the identified trainers testifying their willingness to be part of the experiment. Most importantly, throughout the lecture and demonstration, the trainees contributed their inputs and suggestions for the improvement of mClassRecord.

Launching and End-User Training Outcomes. The end-user training was evaluated excellent regarding the participants' levels of reaction, learning, behavior, results, and management. The quantitative result implies that the trainees are extremely satisfied and engaged during the training, and they see the relevance of the training to their teaching profession. The result of the level of learning connotes that trainees highly perceive that they extremely acquire the intended knowledge, skills, attitudes, confidence, and commitment based on their participation in the end-user training. The result of the level of behavior signifies that the trainees highly believe that they can apply what they learned from the training when they are back on their teaching job. The level of results indicates that the trainees highly believe that targeted outcomes occur as results of the training event and subsequent reinforcement. Moreover, the degree of training management indicates that the trainees highly believe that the training was very organized and managed well.

Usability Result. Table III shows the usability level of mClassRecord. As shown, the mean of all means of the usability level of mClassRecord is 3.61, described as 'strongly agree'. Surprisingly, all usability components—effectiveness ($\bar{x}$ = 3.68), efficiency ($\bar{x}$ = 3.62), satisfaction ($\bar{x}$ = 3.75), learnability ($\bar{x}$ = 3.74), memorability ($\bar{x}$ = 3.65), error-handling ($\bar{x}$ = 3.50), and cognitive load ($\bar{x}$ = 3.32)—were rated with a description of 'strongly agree'.

TABLE III. Usability level.

Usability Components	Mean	Description
Effectiveness	3.68	Strongly agree
Efficiency	3.62	Strongly agree
Satisfaction	3.75	Strongly agree
Learnability	3.74	Strongly agree
Memorability	3.65	Strongly agree
Error-handling	3.50	Strongly agree
Cognitive load	3.32	Strongly agree
Mean of Means	3.61	Strongly agree

The result implies that the respondents can thoroughly complete specified tasks in mClassRecord. It may mean that the teachers can complete the task in mClassRecord with speed and accuracy. The satisfaction result indicates that the teacher-users are comfortable in using mClassRecord. The learnability result means that the technical operation of mClassRecord is easy to learn, and it is easy to operate. The memorability result denotes that the process of using mClassRecord is easy to remember. The error-handling result shows that the respondents can recover easily from errors. Moreover, mClassRecord has a high cognitive load characteristic.

IV. Conclusion

The development of the mobile class record application is a success. Integrating many characteristics of a class record in one app is a challenging application development task. The various class recording practices among different schools in the country affect the comprehensiveness of the app. The app – mClassRecord – is a stand-alone mobile application that automates the manual process of a class information recording and management. It only runs on Android, a Linux-based operating system for mobile devices such as Smartphones and tablet computers. mClassRecord was developed to run on the Jellybean 4.2 version of Android. Just like any other software, it runs on higher, newer versions of the OS, not on lower, older versions. It can be used for all levels of education and all types of learning in the Philippines.

V. Recommendation

The study recommends that mClassRecord will undergo full system testing with live data from the intended users in their classes for one to two or three semesters. This suggestion is to test the robustness of the application. Users must take full responsibility in ensuring the security of the files most especially during hardware crashes or malfunction. It is important to have a backup file of the app and export data to a spreadsheet from time to time. For future development, a multi-platform version of mClassRecord is recommended. It is also recommended to make mClassRecord online and integrate data-syncing feature. Further, the study also recommends incorporating an expert system that would suggest grades based on previous performance and analyze percentage rating that a student needs to pass the subject. Also, there must be regular user training activities that will emphasize on the classroom management and pedagogical use of mClassRecord. Moreover, a feasibility study on its applicability in other countries should be conducted.

Acknowledgment

The funding support provided by the Commission on Higher Education through the Philippine Higher Education Research Network, facilitated by Silliman University through Research and Development Center is gratefully acknowledged. The assistance of Alfie Arcelo is also appreciated. Further, we would like to thank the participants of the 2016 Workshop on Computation: Theory and Practice held at the University of the Philippines – Cebu on September 21-22, 2016 for their comments and suggestions.

References

[1] R. Oberleitner and G. Abowd, "The Importance of Record Keeping," 6 April 2007. [Online]. Available: https://iancommunity.org/cs/articles/the_importance_of_record_keeping.
[2] PTLLSresource.co.uk, "Need for keeping records and types of records," 26 January 2011. [Online]. Available: http://ptllsresource.co.uk/resources/need-for-keeping-records-types-records.
[3] S. Springer, B. Alexander and K. Persiani-Becker, "Teacher Record Keeping," McGraw-Hill Professional, 6 October 2011. [Online]. Available: http://www.education.com/reference/article/teacher-/. [Accessed 28 February 2016].
[4] E. Griffith, "Back to School: The Top Apps for Teachers," *PCMag.com*, 21 August 2013.

[5] PBS LearningMedia, "PBS Survey Finds Teachers are Embracing Digital Resources to Propel Student Learning," *PBS*, 4 February 2013.
[6] D. E. Marcial, "Teacher Education Perceptions of a Proposed Mobile Classroom Manager," *The IAFOR Journal of Education*, pp. 12-28, 2015.
[7] C. L. Sta. Romana, R. Gamboa, D. E. Marcial, G. V. Gabison and A. Sioson, Undergraduate Research and Capstone Project Manual, Philippine Society for Information Technology Educators, 2012.
[8] P. Eeles, "Capturing Architectural Requirements," 15 November 2005. [Online]. Available: http://www.ibm.com/developerworks/rational/library/4706.html#footnotes. [Accessed 20 August 2015].
[9] D. E. Marcial, "Eliciting User Requirements: A Critical Input to the Design of a Mobile Class Record Application," in *13th National Conference on Information Technology Education*, Angeles City, Philippines, 2015.
[10] S. W. Ambler, "Feature Driven Development (FDD) and Agile Modeling," 10 January 2005. [Online]. Available: http://www.agilemodeling.com/essays/fdd.htm. [Accessed 10 January 2016].
[11] S. Goyal, "Feature Driven Development," 2007. [Online]. Available:
[12] U. Eriksson, "How to develop a template for test cases," 28 September 2012. [Online]. Available: http://reqtest.com/tutorials/how-to-develop-a-template-for-test-cases/. [Accessed 20 August 2015].
[13] J. A. Hoffer, J. F. George and J. S. Valacich, Modern Systems Analysis and Design, Upper Saddle River, New Jersey: Pearson Education, Inc, 2011.
[14] G. Wickes, "5 Keys to End-User Training," n. d..
[15] R. Harrison, D. Flood and D. Duce, "Usability of mobile applications: literature review and rationale for a new usability model," *Journal of Interaction Science*, vol. 1, no. 1, pp. 1-16, 2013.
[16] D. Marcial, J. Rendal, P. dela Rama, M. Fortich, J. Etcuban, D. Calibo, "ICT in Teacher Education in Region 7: Benchmarking, Training, and Technology Development towards Innovative Teaching & Learning – Year 1", Technical Report. Silliman University, Philippines.