International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 2 Issue 1, January – 2018, Pages: 1-14

An Intelligent Tutoring System for Learning Android Applications UI Development

Hazem Awni Al Rekhawi and Samy S. Abu Naser
Department of Information Technology, Faculty of Engineering and Information Technology, Al-Azhar University
Gaza, Palestine
Email: *abunaser@alazhar.edu.ps*

Abstract: *The paper describes the design of a web based intelligent tutoring system for teaching Android Applications Development to students to overcome the difficulties they face. The basic idea of this system is a systematic introduction into the concept of Android Application Development. The system presents the topic of Android Application Development and administers automatically generated problems for the students to solve. The system is automatically adapted at run time to the student's individual growth. The system provides obvious support for adaptive demonstration constructs. An initial assessment study was done to examine the effect of using the intelligent tutoring system on the performance of students enrolled in Smartphone Applications Development in the University College of Applied Sciences, Gaza. The results showed a positive impact on the evaluators.*

Keywords: *Intelligent Tutoring System, Android, Java, Problem Generation, Droid-Tutor.*

1. INTRODUCTION

Intelligent Tutoring Systems (ITSs) can be traced back to the early 1970s, when Carbonell tried to combine methods of Artificial Intelligence (AI) with Computer Aided Instruction (CAI) [9]. Therefore, the first generation of ITSs is more or less a sort of "intelligent" CAI. Their focal task is specified by Lelouche: "The basic principle of 'intelligent' CAI is that it should identify the taught material" [15]. Knowledge about the trained material is implanted in the ITS in form of expert systems, that is, the expert module [3,11]. The integration of insights of cognitive science in ITSs, has led to what nowadays is termed an Intelligent Tutoring System [2]. In addition to the knowledge about the trained material, these systems have understanding about pedagogical approaches and understanding about the student, realized as pedagogical module and student module, respectively.

The traditional ITS architecture was designated by Clancey, comprises of the components: student module, expert module, pedagogical module, and user interface [10]. The identification of the components varies. Occasionally, depending on the teaching domain, a module for automatic generation of exercises is likewise part of the ITS. While the ITS's ingredients appear to be part of common agreement in the ITS community, the role and the functionality of each of the components varies a lot [1,3]. The reason for this can partly be seen in the different application domains. One can easily imagine that training in mathematics places different demands on ITSs than clinical medicine training. Another reason might be the realization of different learning theories in ITS. Case-based learning, described in the former section, places special demands on an ITS that are rather different in problem-oriented education. These sensible features often unavoidably lead to mixed and unrivaled systems. But even in the same application domain and established on the same education style, ITSs are often not analogous and based on a whole different clarification of the same architecture. Furthermore, regarding only the ITS architecture on a more nonconcrete level, it turn out to be tough to find motives for heterogeneous understandings at all. A mixture of content and delivery functions, which is seemingly not based on visions of research but on traditions of ITS development, can be found in ITS understandings.

In other ITSs, the expert knowledge base is a modest database deprived of own functionality [17]. The same condition can be found in the dissimilar ways the pedagogical knowledge module is realized and embedded in the ITS. Thus, there are ITSs that involve a set of interacting and more or less distinct subsystems, and there are ITSs containing of passive components plus a component that encapsulates the execution. Execution in this context contains the interaction with the student, the evaluation of the student's behavior and success, and the provision of contents and navigation. Thus, two standpoints on the same architecture can be established, replicating different clarification of the same modules: ITS architecture contains either distinct independent subsystems or reflexive components with unified implementation system.

Both viewpoints have their benefits and drawbacks. However, the main system's thinking concerning the understanding of the components should be made clear to offer comparability of ITSs and reusability of ITS components. The benefit of this approach is that the central navigating component might be reused in different ITSs, as it is clearly detached from the databases and the user interface. Figure 1 illustrate the recommended ITS architecture with the tutoring process module is outlined.

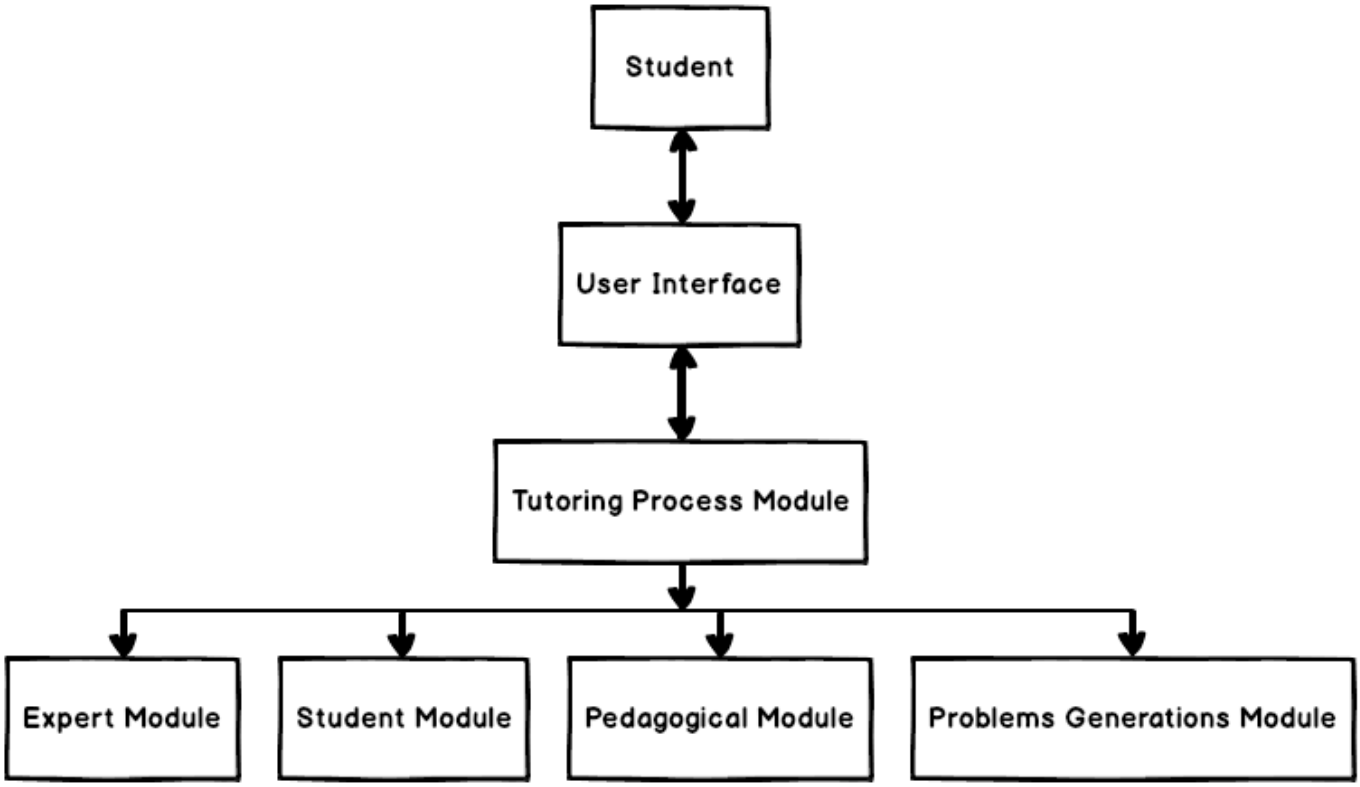

Figure 1. ITS architecture with tutoring process module.

The design of the Intelligent Tutoring System for Learning Android Applications Development (Droid-Tutor) adopted the ITS architecture with automatically generated problems module.

The benefits of the Droid-Tutor are:

- Building an intelligent tutoring system for problems for which the responses might not be usually quantitative.
- Creating limitless number of problems automatically, thereby providing as much practice with problem-solving as the student wants.
- Having a system that adapt at run time to the student's individual growth.
- Using students' previous knowledge of Java in the development of android applications.
- Teaching students how to deal with problems and solve them.
- Evaluating student level through MCQ, correct and error on several levels of difficulty.
- Providing a complete teaching material for the student in android applications development containing a lot of exercises and practical examples.
- Providing videos that explain some lessons in Android applications development, making learning easier for students.
- Designing of Android applications according to the standards of mobile applications design.
- Building interactive android applications and publishing it on Google Play Store.

2. DROID-TUTOR DESIGN

There are many Intelligent Tutoring Systems (ITS) that were designed and developed for many fields such as: a comparative study between Animated Intelligent Tutoring Systems (AITS) and Video-based Intelligent Tutoring

Systems (VITS) [37], cloud computing[34], Computer Networks [52], ADO-Tutor: Intelligent Tutoring System for leaning ADO. NET [53],CPP-Tutor for C++ Programming Language [30], effectiveness of E-learning [30, 36], effectiveness of the CPP-Tutor [57], E-learning system [24], Health problems related to addiction of video game playing [47], Java Expression Evaluation [36] , Linear Programming [42], learning Computer Theory [60], Learning Java Objects [3], Mathematics Intelligent Tutoring System [43], Oracle Intelligent Tutoring System (OITS) [48], parameter passing in Java programming [35], Photoshop (CS6) Intelligent Tutoring System [59], predicting learners performance using NT and ITS [41], stomach disease Intelligent Tutoring System [52], teaching advanced topics in information security [57], teaching AI searching algorithms [40], teaching database [38], teaching DES information security algorithm[55], teaching grammar English tenses [49], teaching Mongo Database [56], teaching the 7 Characteristics for Living Things [59], right letter pronunciation in reciting the Holy Quran [44] , and Diabetes Intelligent Tutoring System [51], CSS-Tutor: An intelligent tutoring system for CSS and HTML[45], Design and Development of an Intelligent Tutoring System for C#[46].

3. DROID-TUTOR IS UNIQUE WITH RESPECT TO THE PREVIOUS WORK IN THE FOLLOWING MANNER:

- The system was built to look like the most familiar interactive interfaces like OS desktops, including (icons, drag and drop features, tabs menus, and pop-up windows) which are all integrated in one single window.
- The intelligent tutoring system was built for problems for which the answers might not be always quantitative [1,2].
- WebToTeach is related to Droid-Tutor [5], but it administers instructor pre-prepared problems, and does not generate the problems automatically. Droid-Tutor can generate unlimited number of problems automatically, thereby providing as much practice with problem-solving as the student needs.

Kashy have shown that the use of problem generation systems has increased student performance by 10% in Physics, largely due to limitless time spent on the task [13].

4. DROID-TUTOR IS DESIGNED TO HELP STUDENTS LEARN ANDROID APPLICATIONS DEVELOPMENT BY:

1. Gradually teaching the Android Applications Development material to the students. The system is supported with a student controlled voice narrator, which acts as a facility during learning.
2. Repeatedly solving automatically generated problems and obtaining the proper feedback.
3. Droid Tutor is designed with amazing android text lessons and video training materials.
4. The possibility of measuring the level of the student through exercises and questions.
5. The possibility of advising the student how to handle errors and solve them quickly.

The Droid-Tutor is designed to be used as a supplement to the traditional method of teaching (Android Development textbook and Instructor), either during a laboratory session, after class training, or for homework assignments.
Droid-Tutor has the following modules: Pedagogical Module, Problems Generation Module, and Expert Module, Student Module, and Tutoring process module.

4.1 Droid-Tutor Pedagogical Module Design

It has been noticed that students are having difficulties in understanding the concepts of Android Applications Development. To overcome these difficulties, an Intelligent Tutoring System for teaching Android Applications Development called Droid-Tutor have been developed to students enrolled in Android Applications Development Course in the Mobile Applications Development Department at University College of Applied Sciences in Gaza. Droid-Tutor gradually introduces students to the concept of Android Applications Development and automatically generates problems for the students to solve. The key sections that draws the main structure of the tutoring material [18,19,20] are:

4.2 Android Overview and history

- Mobile phones revolution
- What is Android? Android Features, Android Versions
- Android Devices on the market
- Android Architecture
- The Advantage of adopting Android as a Developer
- Android Applications

- Android Development Environment
- Android Development Prerequisites

4.3 Basics Android User Interface

- Android Applications Components
- Installing Android Studio Environment
- Create First Android Application and Run AVD
- Anatomy of Android Application
- Layout Manager and Android Views
- Android UI Control
- Activity Lifecycle
- Views Events Handling
- Build Styles and Themes
- Dealing with Container Views and Adapters
- Build Menu Groups and Items
- Build Android Dialogs
- Intent and Intent Filters

4.4 Advanced Android Applications Design

- Android Fragments and Fragment Manager
- Android Drawable Styles
- Android Animations
- Android Material Designs
- Advanced UI Examples
- Localizations and Multilanguage's
- Support Multiscreen mobile and tablet

4.5 Problems Generation Module of Droid-Tutor

A number of problems have been lately recognized as a potential shortcoming of encoding a limited set of problems into a tutor [16]. A technique used in literature to dynamically generate problems is by BNF-like grammar [14]. In this technique, problems are generated by arbitrarily instantiating templates written in the grammar. Each template can be prudently planned with specific pedagogical aims in mind.

Problem Generation Module is accountable for generating the template of code, theses templates cover the core topics of Android Applications Development which are (android user interface, applications components, Activity Lifecycle, Layout Manager, Android views, Android Dialogs, Fragments and Advanced Android Material Design). The module depends on randomly structuring of the pieces of codes which forms the templates that consist of (layout manager, values resource, themes and styles, Custom Data Adapter, return types, arguments data types, classes, methods, and arguments names). These structures are imported from pre-defined lists of keywords.
The template is generated with a previously intended problem each time it is requested, followed by a related question and possible solutions. The questions have different styles including either asking the student to correct an Android code, write an Android code, multiple choice, or true/false.

4.6 Expert Module of Droid-Tutor

Expert Module was implemented to gather the necessary information for generating the feedback [4]. The expert module is capable of solving the generated problems by parsing the template. Since the expert module can execute any code, it can generate the correct answer for a problem on its own, and determine whether the user's answer is correct/incorrect. In addition to whether the user's answer is correct/incorrect, the module can provide the student with the correct answer when it is requested. Furthermore, the module provides the student the proper feedback in response to the student's answer.

4.7 Student Module of Droid-Tutor

A new student must create his own account to have a profile. The profile has information about the student such as his name, dates of login, score of each session, and learning progress during each session. The student's score can be

viewed at any time during the session that describes the student performance in solving problems in the following subjects: casting, classes and inheritance.

4.8 Tutoring process Module of Droid-Tutor

Tutoring process module works as a coordinator that controls the functionality of the whole system.

4.9 Droid-Tutor User Interface Design

The Droid Tutor is uniquely designed and easy for the student and here is a screenshot and explanation for each interface, you can see figures from Figure 2 to Figure 14.

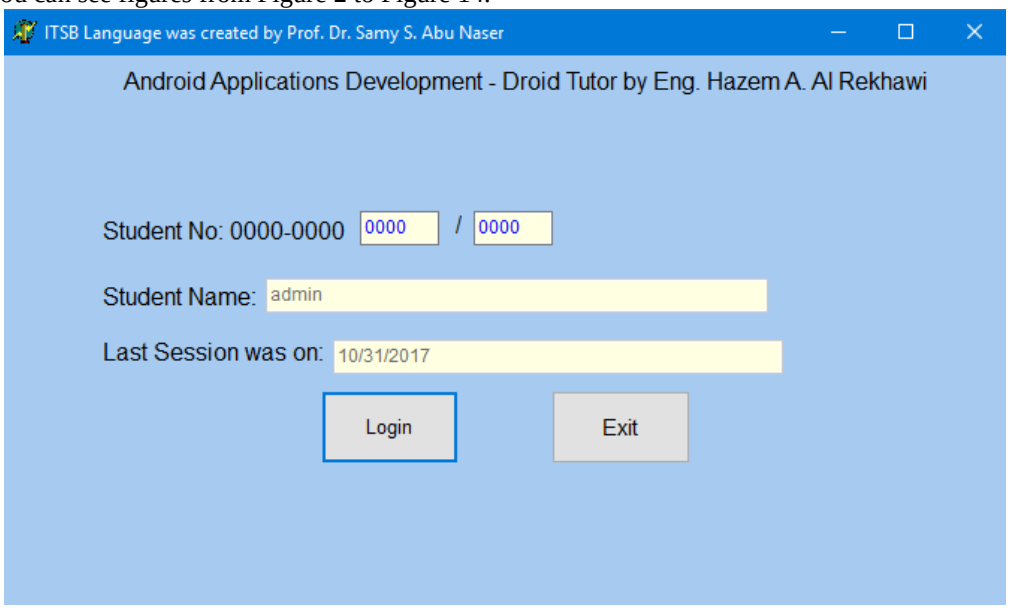

Figure 2. The Login User Interface.

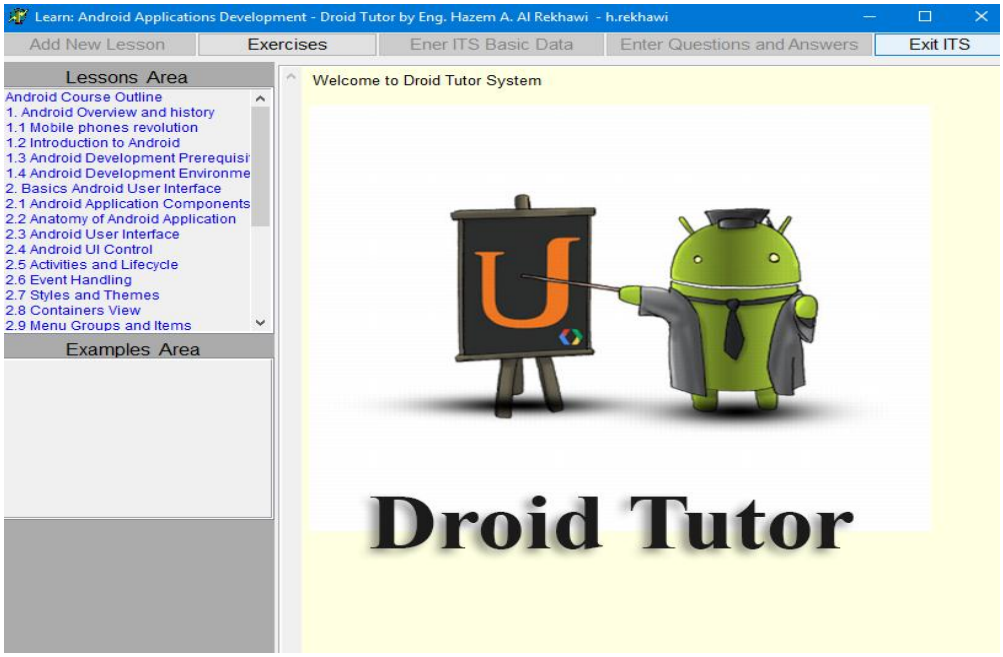

Figure 3. The Main User Interface.

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 2 Issue 1, January – 2018, Pages: 1-14

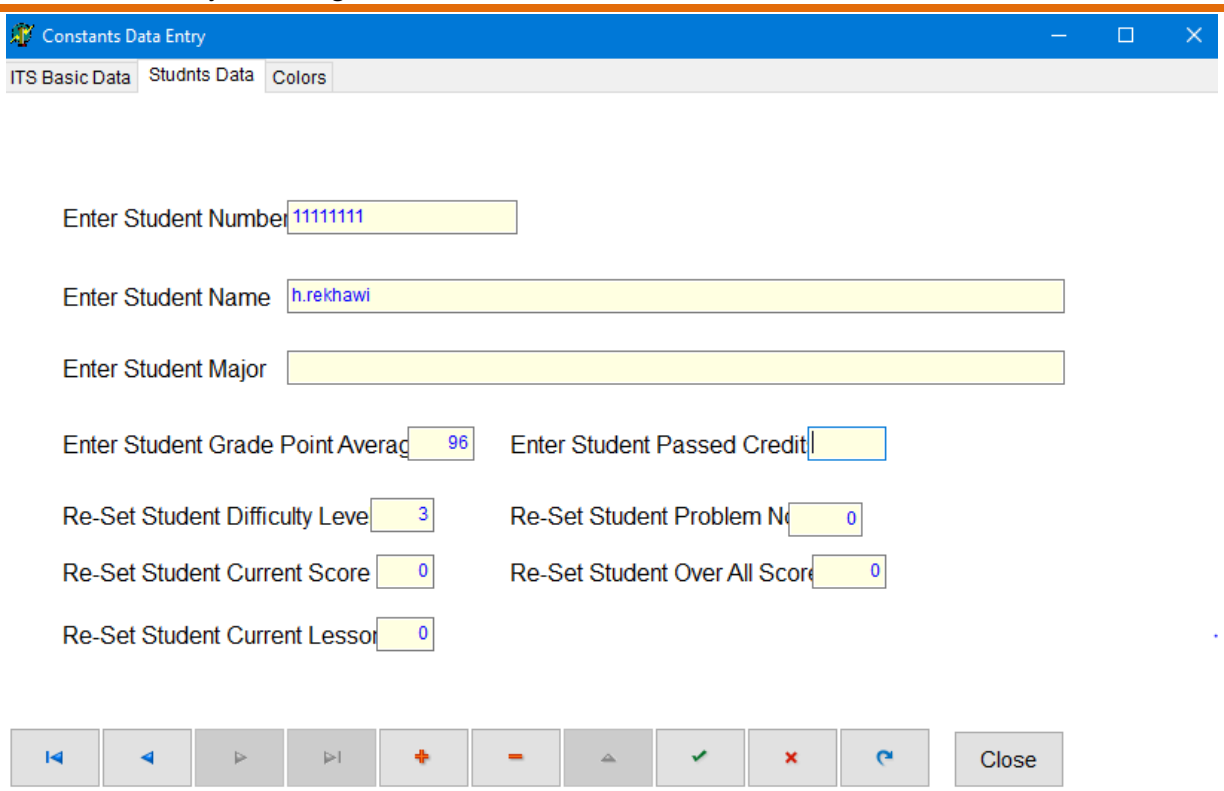

Figure 4. Student Data Interface.

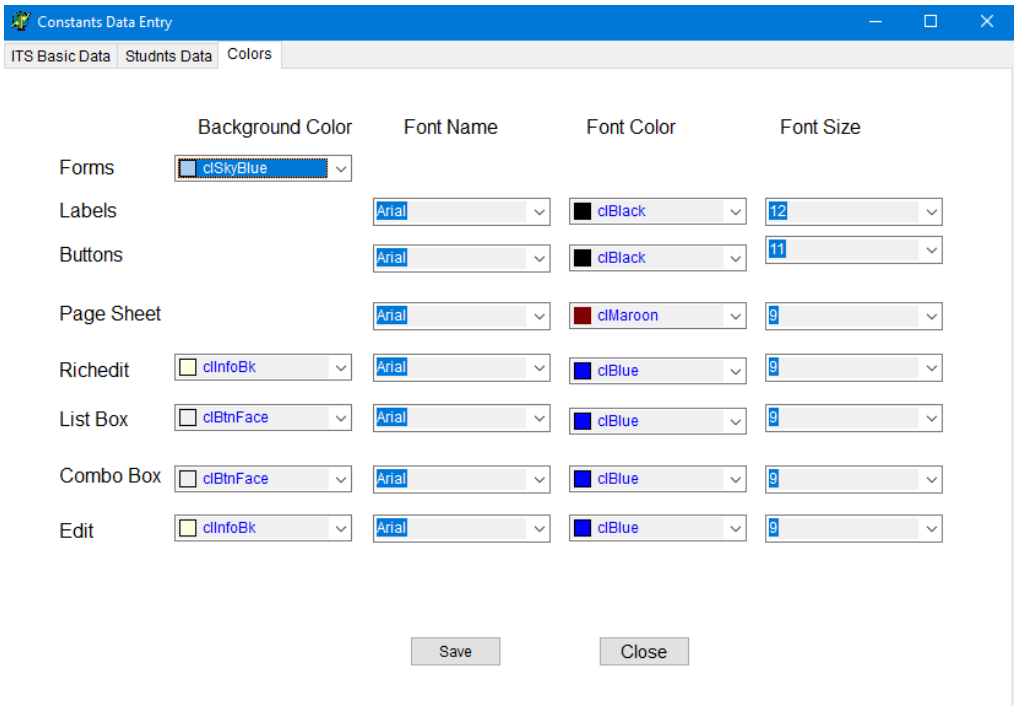

Figure 5. Controls Fonts and Colors User Interface.

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 2 Issue 1, January – 2018, Pages: 1-14

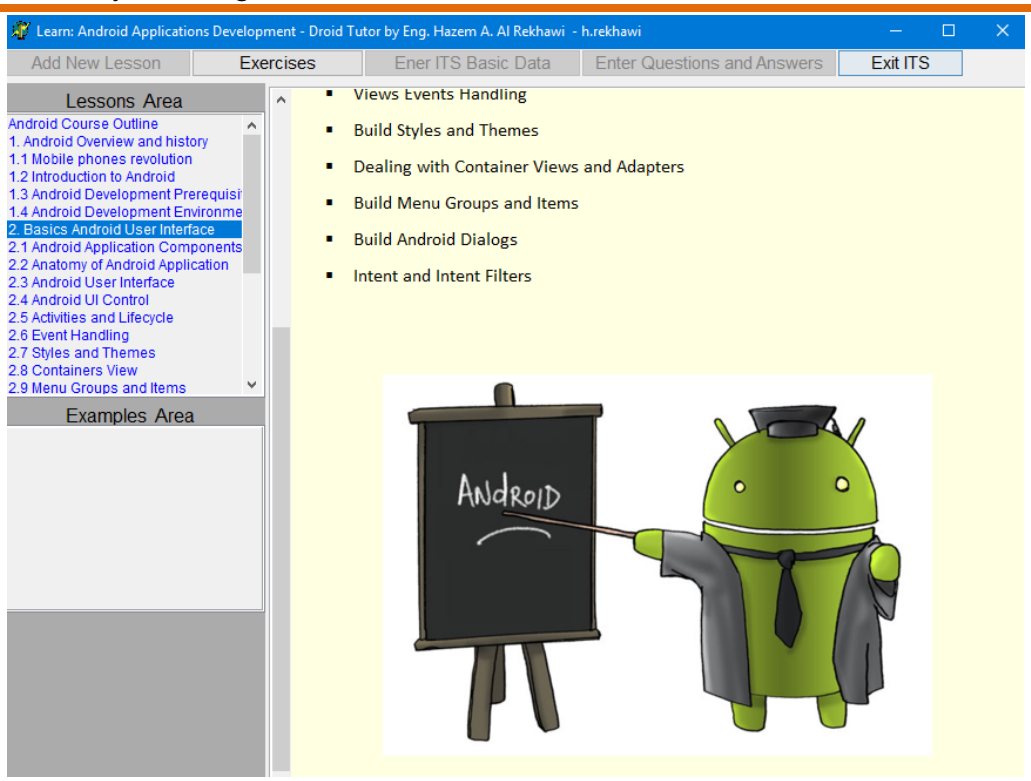

Figure 6. Learning Material Android Applications Development Screenshot 1.

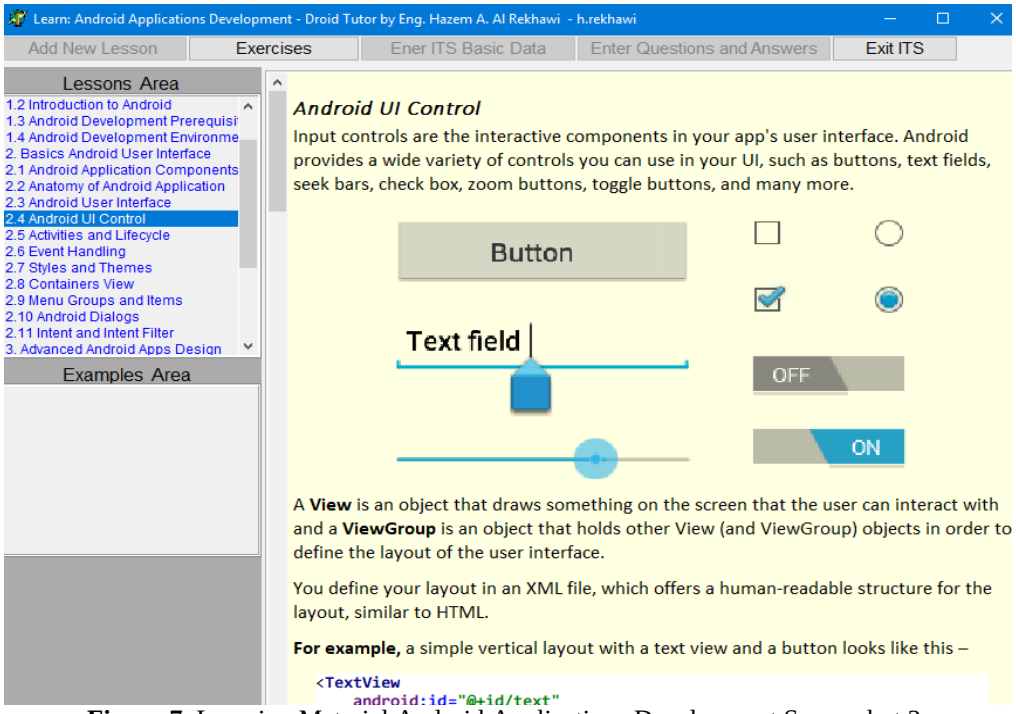

Figure 7. Learning Material Android Applications Development Screenshot 2.

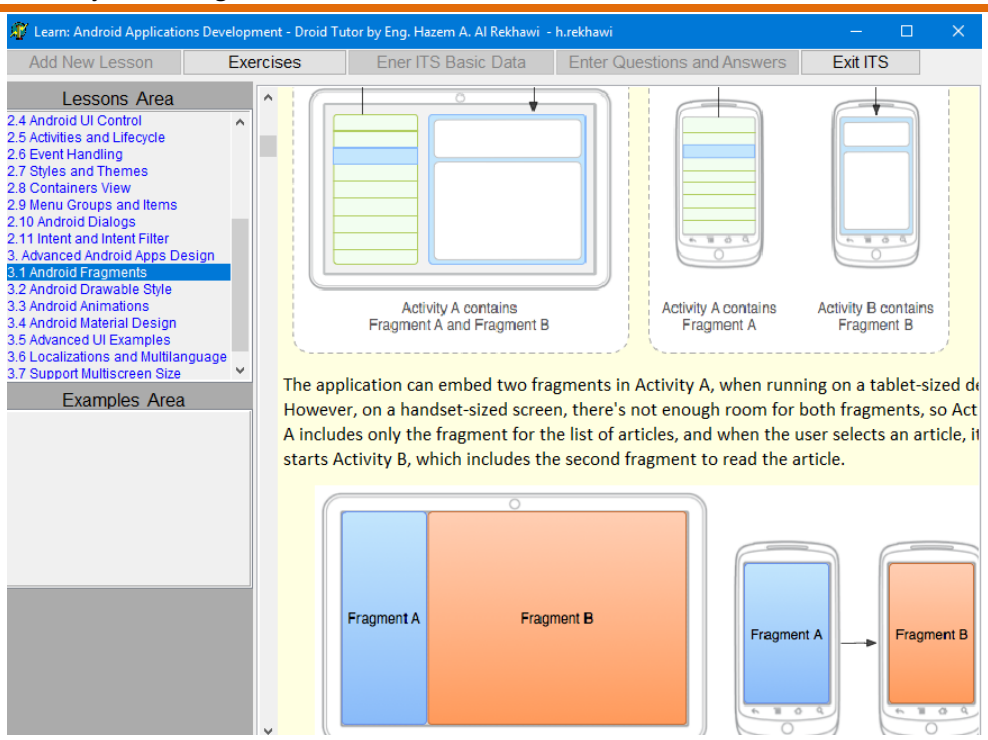

Figure 8. Learning Material Android Applications Development Screenshot 3.

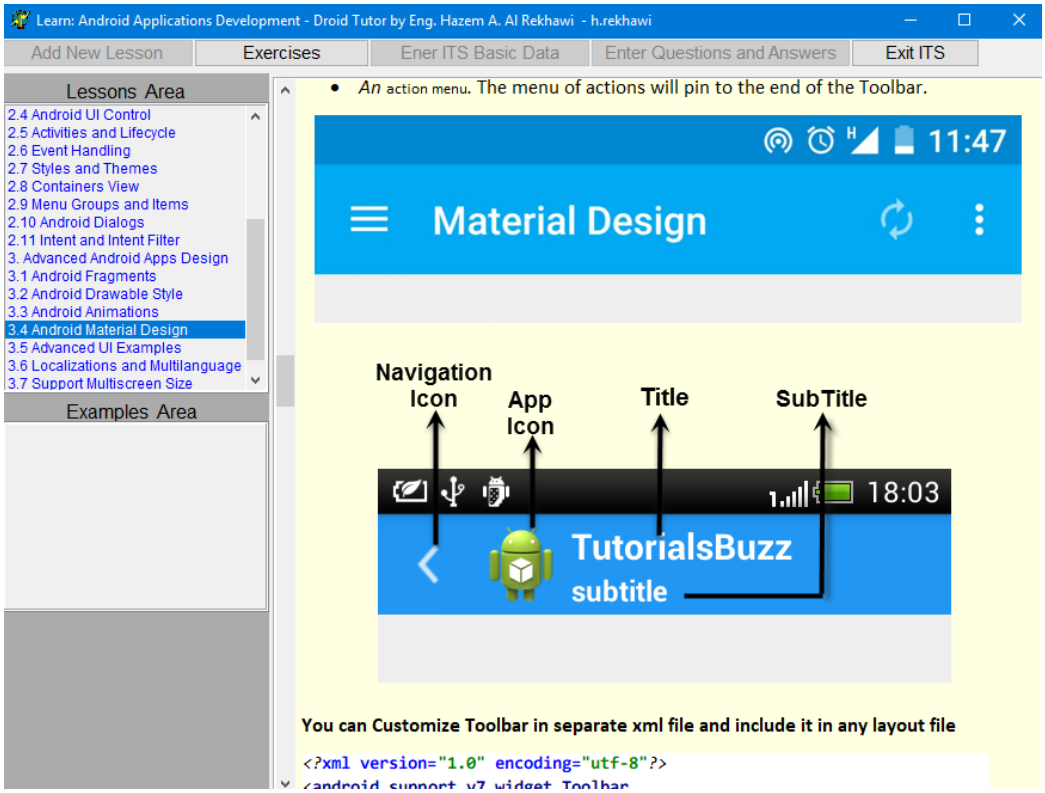

Figure 9. Learning Material Android Applications Development Screenshot 4.

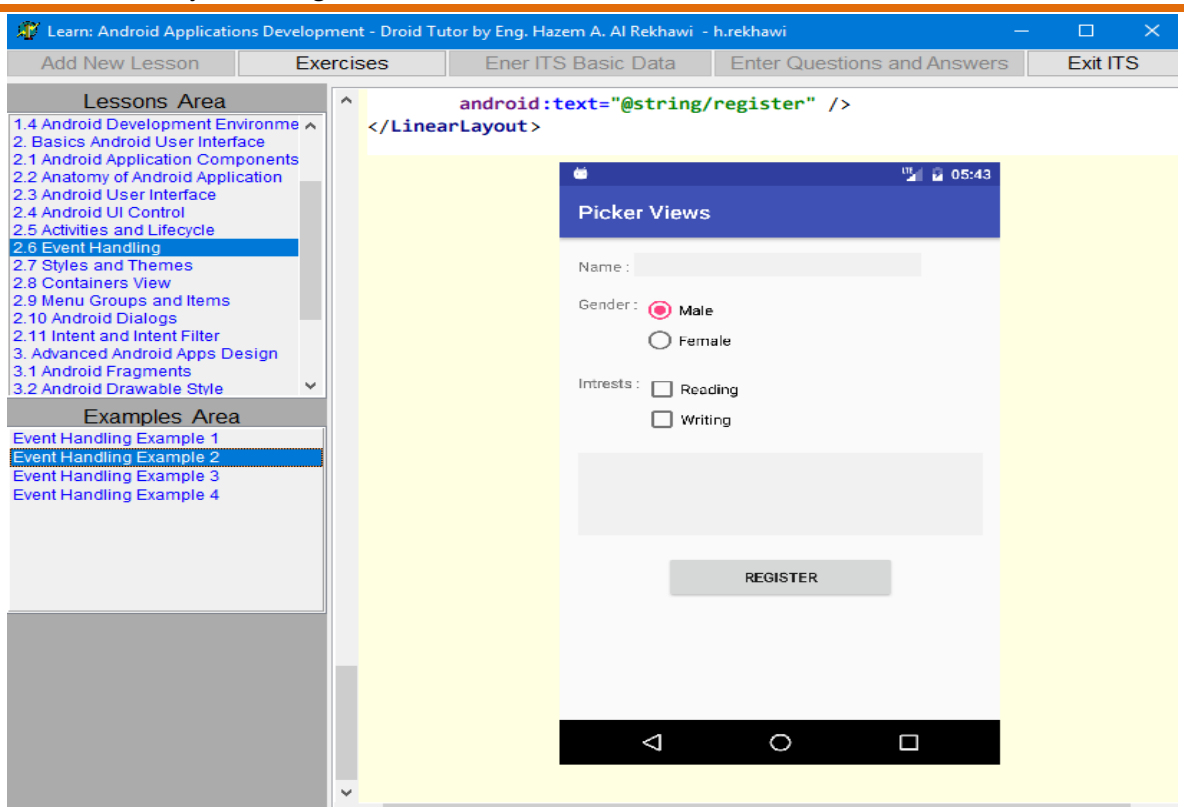

Figure 10. Examples of Lessons User Interface.

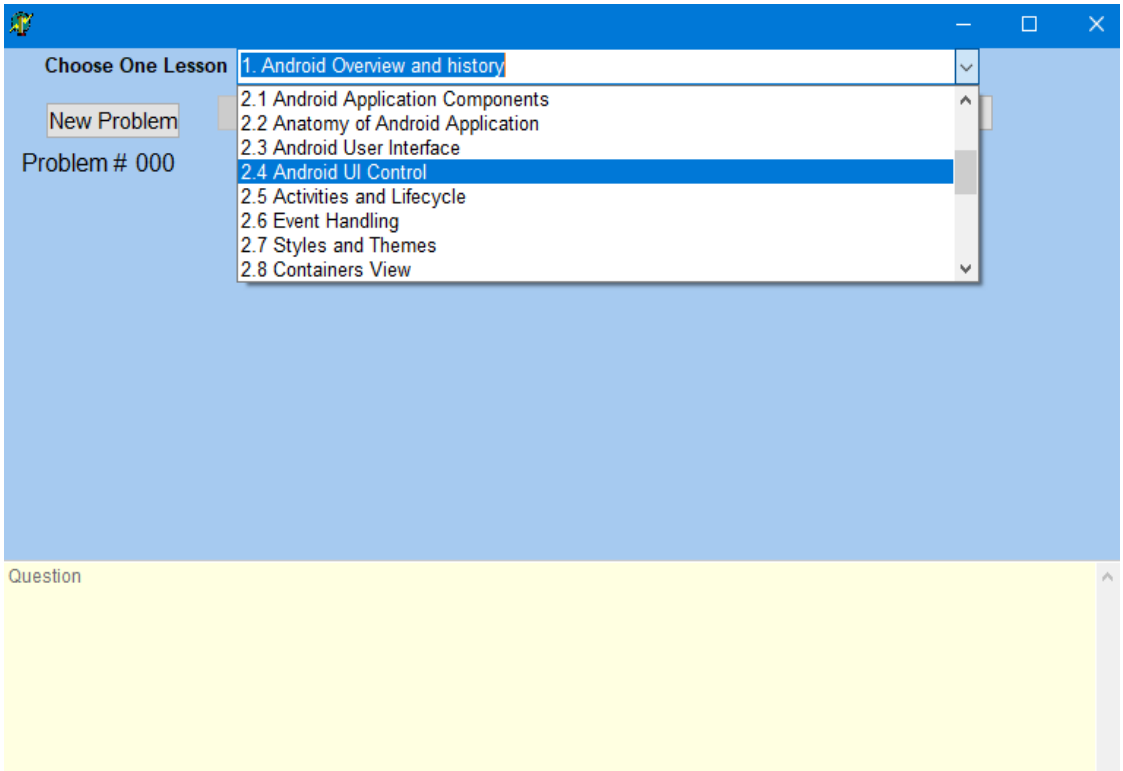

Figure 11. Exercises and Questions User Interface.

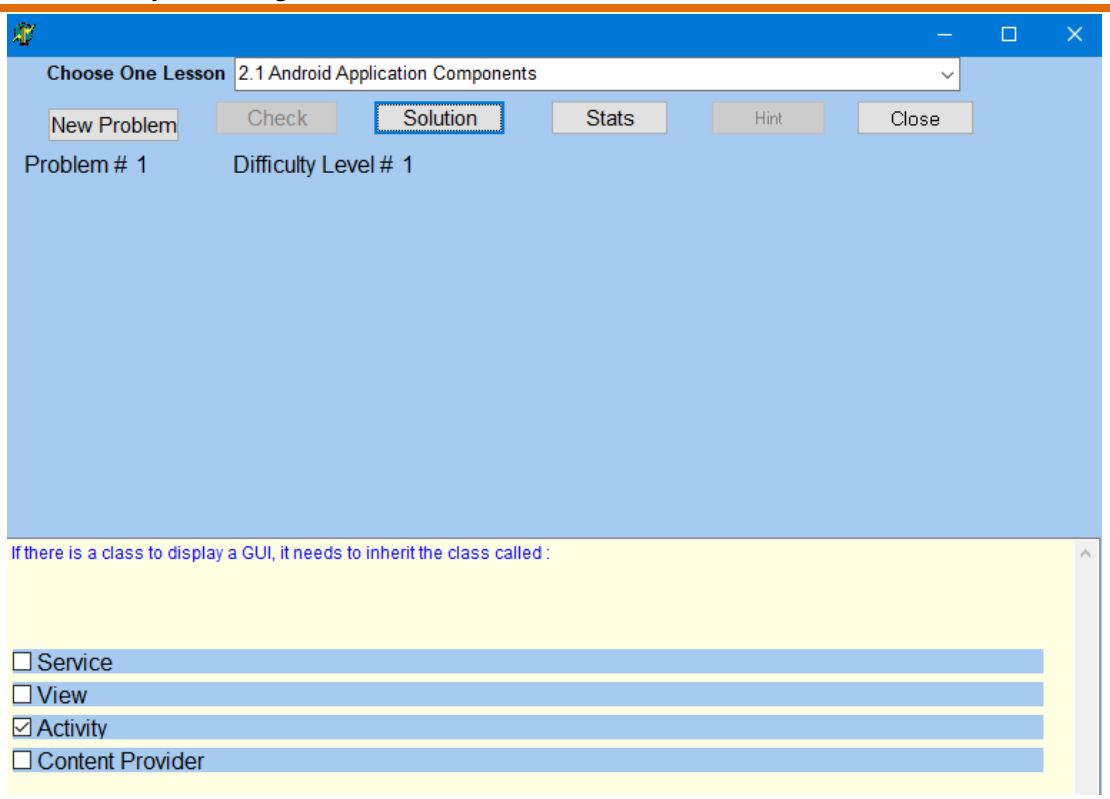

Figure 12. Different Types of Questions – Multiple Choice.

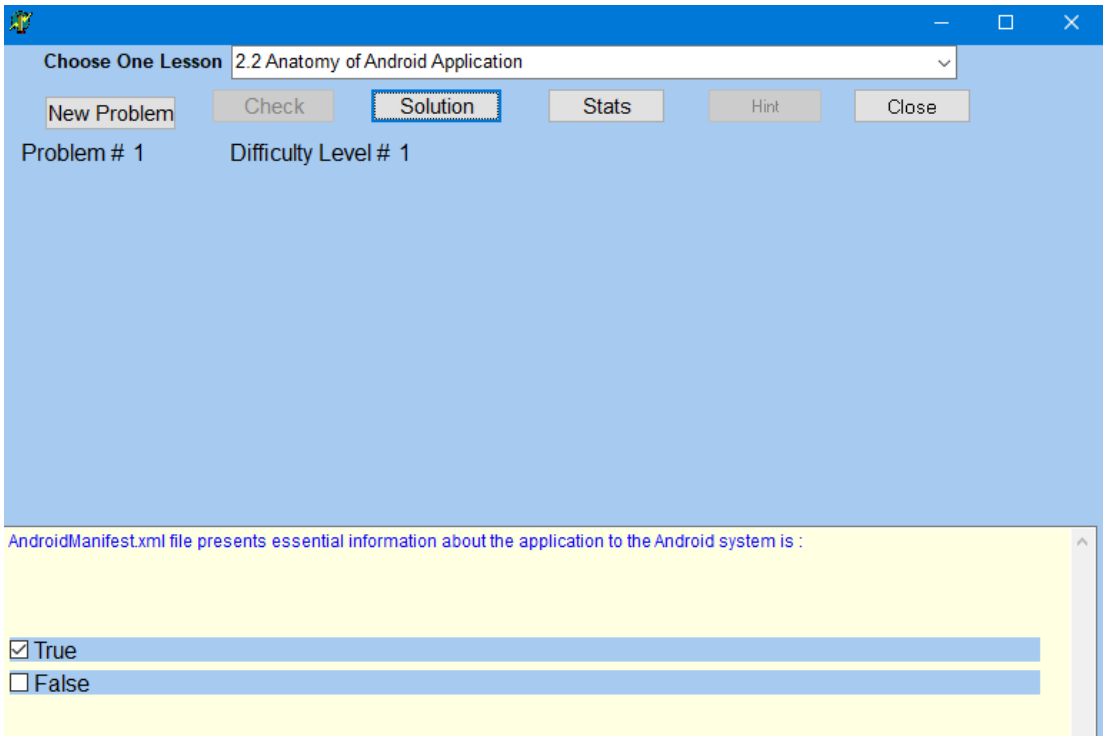

Figure 13. Different Types of Questions – True or False.

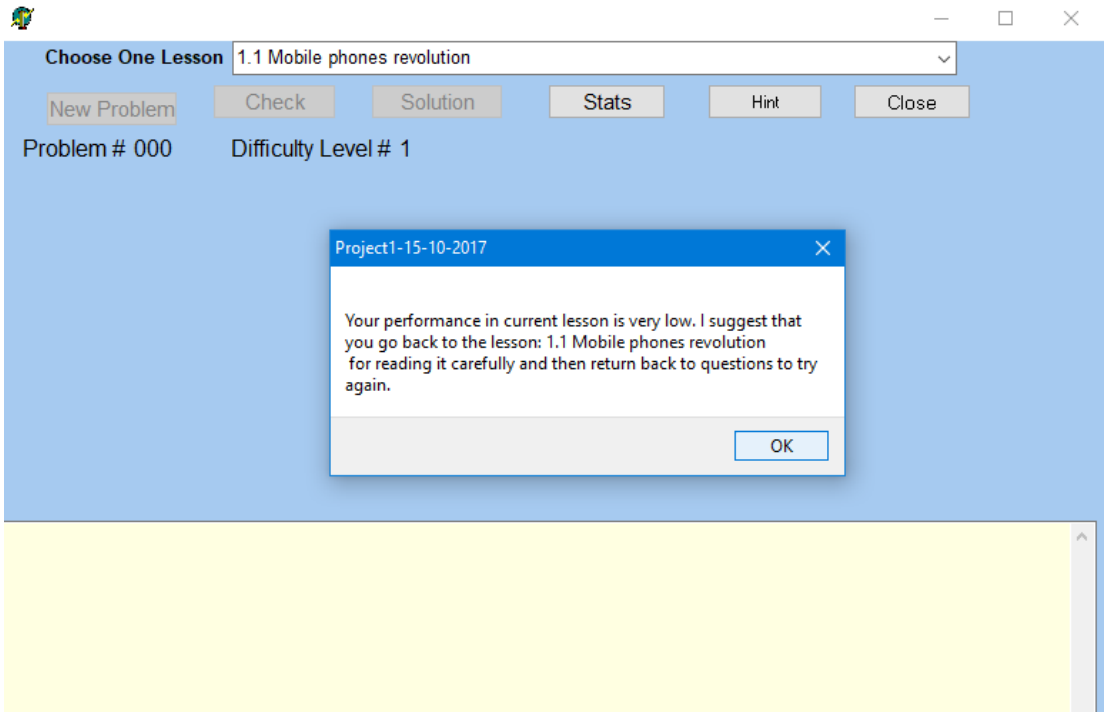

Figure 14. Result of Final Evaluation.

5. EVALUATION OF THE DROID-TUTOR

An initial evaluation of the Droid-tutor was carried out by the lecturers and their students who enrolled in android applications development course during the first semester of 2017/2018 in Mobile Applications Development Department at University College of Applied Sciences in Gaza. A questioner consisting of the items in table 1 was filled out by each evaluator (Lecturers and Students). A group of 4 lecturers and 25 students participated in the evaluation of the system. Table 1 shows the overall rating of the lecturers and students who evaluated the system.

Table 1: Shows the rating of the of the Droid-Tutor by lecturers and students

Item #	Item	Rating %	
		Lecturers	Students
1	The Quality of the Droid-Tour Design	94%	96%
2	The importance of the topic covered (Android Applications Development)	96%	98%
3	Would you benefit from using the Droid-Tutor?	92%	98%
4	Do you recommend using Droid-Tutor for android applications development course as a supportive tool?	100%	100%
5	Would you like to see similar tutoring system in other courses?	100%	100%

Form the summary of Table 1, the evaluation of the Droid-Tutor showed a positive impact on the evaluators (Lecturers and students). Furthermore, they recommend that similar systems for other courses to be implemented.

6. CONCLUSIONS AND FUTURE WORKS

The design of an Intelligent Tutoring System called Droid-Tutor was described in this paper. Droid-Tutor was designed for teaching Android Applications Development to students to overcome their difficulties. Droid Tutor

presents the topic of Android Applications Development to the student and administers automatically generated problems for him to solve. Droid-Tutor is dynamically adapted at run time to the student's individual progress. An initial evaluation of Droid-Tutor was carried out by the lecturers and students taken the Android Applications Development course in the University College of Applied Sciences in Gaza. The outcome of the evaluation was positive and suggested that other intelligent tutoring systems be designed for other courses. We recommend a comprehensive evaluation of the system to be carried out next time the course is offered.

REFERENCES

[1] Al Rekhawi, H. A., Ayyad, A. A., & Abu Naser, S. S. (2017). Rickets Expert System Diagnoses and Treatment. International Journal of Engineering and Information Systems (IJEAIS), 1(4), 149-159.
[2] Abu Naser, S. S. (2016). ITSB: An Intelligent Tutoring System Authoring Tool. Journal of Scientific and Engineering Research, 3(5), 63-71.
[3] Amalathas, S., A. Mitrovic, R. Saravanan and Divison, 2010. Developing an intelligent tutoring system for palm oil in ASPIRE. Proceedings of the 18th International Conference on Computers in Education, Nov. 29-Dec. 3, Asia Pacific Society for Computers in Education, Putrajaya, Malaysia, pp: 101-103
[4] Anderson, J.R., A.T. Corbett, K.R. Koedinger and R. Pelletier, 1995. Cognitive tutors: Lessons learned. J. Learn. Sci., 4: 167-207.
[5] Arnow, D. and O. Barshay, 1999. WebToTeach: An interactive focused programming exercise system. Proc. Annu. Frontiers Educ. Conf., 1: 12A9/39 - 12A9/44
[6] Barker, D.S., 1997. CHARLIE: A computer-managed homework, assignment and response, learning and instruction environment. Proc. Annu. Frontiers Educ. Conf. Teach. Learn. Era Change, 3: 1503 – 1509
[7] Bridgeman, S., M.T. Goodrich, S.G. Kobourov and R. Tamassia, 2000. PILOT: An interactive tool for learning and grading. Proc. SIGCSE Tech. Symp. Comput. Sci. Educ., 32: 139-143.
[8] Bridgeman, S., M.T. Goodrich, S.G. Kobourov and R. Tamassia, 2001. SAIL: A system for generating, archiving and retrieving specialized assignments using LaTeX. Proceedings of the 31st SIGCSE Technical Symposium, Mar. 7-12, Austin, TX, pp: 300-304
[9] Carbonell, J., 1970. AI in CAI: An artificial-intelligence approach to computerassisted instruction. IEEE Trans. Man Mach. Syst., 11: 190 - 202
[10] Clancey, W., 1984. Methodology for Building an Intelligent Tutoring System. In: Methods and Tactics in Cognitive Sciences, Kintsch, W., J.R. Miller and P.G. Polson (Eds.). Lawrence Erlbaum Associates, Hillsdale, New Jersey, London, pp: 51-84
[11] Fournier-Viger, P., R. Nkambou and E. Mephu, 2010. Building Intelligent Tutoring Systems for Ill-Defined Domains. In: Advances in Intelligent Tutoring Systems, Nkambou, R., R. Mizoguchi, and J. Bourdeau (Eds.). SpringerVerlag, Berlin, Heidelberg, pp: 81-101.
[12] Kashy, E., B.M. Sherrill, Y. Tsai, D. Thaler, D. Weinshank, M. Engelmann and D.J. Morrissey, 1993. CAPA-An integrated computer-assisted personalized assignment system. Am. J. Phys., 61: 1124-1130.
[13] Kashy, E., M. Thoennessen, Y. Tsai, N.E. Davis and S.L. Wolfe, 1997. Using networked tools to enhance student success rates in large classes. Proc. Annu. Frontiers Educ. Conf. Teach. Learn. Era Change, 1: 233-237
[14] Koffman, E.B. and J.M. Perry, 1976. A module for generative CAI and concept selection. Int. J. Man Mach. Stud., 8: 397-410.
[15] Lelouche, R., 1999. Intelligent tutoring systems from birth to now. Kunstliche Intelligenz, 13: 5-11.
[16] Martin, B. and A. Mitrovic, 2000. Tailoring feedback by correcting student answers. Proc. Intell. Tutor. Syst., 1839: 383-392.
[17] Tang, Y., 2005. Qualitative reasoning and articulate software. Inform. Technol. J., 4: 184-188.
[18] Android developer site https://developer.android.com/develop/index.html.
[19] Android Applications Development Training Material by Eng. Hazem A. S. Al Rekhawi Expert in Android Applications Development. https://www.linkedin.com/in/hazem-al-rekhawi-0372b358
[20] Android Hive website that is contains a lot of android examples. https://www.androidhive.info/
[21] Android Tutorial Point website. https://www.androidtutorialpoint.com/
[22] Google page on GitHub site. https://github.com/googlesamples
[23] Mobile development category at tuts plus website. https://code.tutsplus.com/categories/mobile-development
[24] Android Development by Vogella Website. http://www.vogella.com/tutorials/android.html
[25] Full Android Development Course by tutorials point website https://www.tutorialspoint.com/android/

[26] Mrouf, A., Albatish, I., Mosa, M., & Abu Naser, S. S. (2017). Knowledge Based System for Long-term Abdominal Pain (Stomach Pain) Diagnosis and Treatment. International Journal of Engineering and Information Systems (IJEAIS), 1(4), 71-88.
[27] AbuEl-Reesh, J. Y., & Abu Naser, S. S. (2017). An Expert System for Diagnosing Shortness of Breath in Infants and Children. International Journal of Engineering and Information Systems (IJEAIS), 1(4), 102-115.
[28] Bakeer, H. M. S., & Naser, S. S. A. (2017). Photo Copier Maintenance Expert System V. 01 Using SL5 Object Language. International Journal of Engineering and Information Systems (IJEAIS), 1(4), 116-124.
[29] El Agha, M., Jarghon, A., & Abu Naser, S. S. (2017). Polymyalgia Rheumatic Expert System. International Journal of Engineering and Information Systems (IJEAIS), 1(4), 125-137.
[30] Khella, R. A., & Abu Naser, S. S. (2017). Expert System for Chest Pain in Infants and Children. International Journal of Engineering and Information Systems (IJEAIS), 1(4), 138-148.
[31] Nabahin, A., Abou Eloun, A., & Abu Naser, S. S. (2017). Expert System for Hair Loss Diagnosis and Treatment. International Journal of Engineering and Information Systems (IJEAIS), 1(4), 160-169.
[32] Abu Ghali, M. J., Mukhaimer, M. N., Abu Yousef, M. K., & Abu Naser, S. S. (2017). Expert System for Problems of Teeth and Gums. International Journal of Engineering and Information Systems (IJEAIS), 1(4), 198-206.
[33] Qwaider, S. R., & Abu Naser, S. S. (2017). Expert System for Diagnosing Ankle Diseases. International Journal of Engineering and Information Systems (IJEAIS), 1(4), 89-101.
[34] Abu Hasanein, H. A., & Abu Naser, S. S. (2017). An intelligent tutoring system for cloud computing. International Journal of Academic Research and Development, 2(1), 76-80.
[35] Abu Naser, S. (2008). An Agent Based Intelligent Tutoring System For Parameter Passing In Java Programming. Journal of Theoretical & Applied Information Technology, 4(7).
[36] Abu Naser, S. (2008). JEE-Tutor: An Intelligent Tutoring System for Java Expression Evaluation. Information Technology Journal, Scialert, 7(3), 528-532.
[37] Abu Naser, S. S. (2001). A comparative study between animated intelligent tutoring systems AITS and video-based intelligent tutoring systems VITS. Al-Aqsa Univ. J, 5(1), 72-96.
[38] Abu Naser, S. S. (2006). Intelligent tutoring system for teaching database to sophomore students in Gaza and its effect on their performance. Information Technology Journal, 5(5), 916-922.
[39] Abu Naser, S. S. (2008). Developing an intelligent tutoring system for students learning to program in C++. Information Technology Journal, 7(7), 1055-1060.
[40] Abu Naser, S. S. (2008). Developing visualization tool for teaching AI searching algorithms. Information Technology Journal, Scialert, 7(2), 350-355.
[41] Abu Naser, S. S. (2012). Predicting learners performance using artificial neural networks in linear programming intelligent tutoring system. International Journal of Artificial Intelligence & Applications, 3(2), 65.
[42] Abu Naser, S. S. (2012). A Qualitative Study of LP-ITS: Linear Programming Intelligent Tutoring System. International Journal of Computer Science & Information Technology, 4(1), 209.
[43] AbuEloun, N. N., & Abu Naser, S. S. (2017). Mathematics intelligent tutoring system. International Journal of Advanced Scientific Research, 2(1), 11-16.
[44] Akkila, A. N., & Abu Naser, S. S. (2017). Teaching the right letter pronunciation in reciting the holy Quran using intelligent tutoring system. International Journal of Advanced Research and Development, 2(1), 64-68.
[45] Alawar, M. W., & Abu Naser, S. S. (2017). CSS-Tutor: An intelligent tutoring system for CSS and HTML. International Journal of Academic Research and Development, 2(1), 94-98.
[46] Al-Bastami, B. G., & Abu Naser, S. S. (2017). Design and Development of an Intelligent Tutoring System for C# Language. EUROPEAN ACADEMIC RESEARCH, 6(10), 87-95.
[47] Al-Bayed, M. H., & Abu Naser, S. S. (2017). An intelligent tutoring system for health problems related to addiction of video game playing. International Journal of Advanced Scientific Research, 2(1), 4-10.
[48] Aldahdooh, R., & Abu Naser, S. S. (2017). Development and Evaluation of the Oracle Intelligent Tutoring System (OITS). EUROPEAN ACADEMIC RESEARCH, 6(10), 8711-8721.
[49] Alhabbash, M. I., Mahdi, A. O., & Abu Naser, S. S. (2016). An Intelligent Tutoring System for Teaching Grammar English Tenses. EUROPEAN ACADEMIC RESEARCH, 6(9), 7743-7757.
[50] Al-Hanjori, M. M., Shaath, M. Z., & Abu Naser, S. S. (2017). Learning computer networks using intelligent tutoring system. International Journal of Advanced Research and Development(2), 1.
[51] Almurshidi, S. H., & Abu Naser, S. S. (2017). Design and Development of Diabetes Intelligent Tutoring System. EUROPEAN ACADEMIC RESEARCH, 6(9), 8117-8128.

[52] Almurshidi, S. H., & Abu Naser, S. S. (2017). Stomach disease intelligent tutoring system. International Journal of Advanced Research and Development, 2(1), 26-30.
[53] El Haddad, I. A., & Abu Naser, S. S. (2017). ADO-Tutor: Intelligent Tutoring System for leaning ADO. NET. EUROPEAN ACADEMIC RESEARCH, 6(10), 8810-8821.
[54] Elnajjar, A. E. A., & Abu Naser, S. S. (2017). DES-Tutor: An Intelligent Tutoring System for Teaching DES Information Security Algorithm. International Journal of Advanced Research and Development, 2(1), 69-73.
[55] Hilles, M. M., & Abu Naser, S. S. (2017). Knowledge-based Intelligent Tutoring System for Teaching Mongo Database. EUROPEAN ACADEMIC RESEARCH, 6(10), 8783-8794.
[56] Mahdi, A. O., Alhabbash, M. I., & Abu Naser, S. S. (2016). An intelligent tutoring system for teaching advanced topics in information security. World Wide Journal of Multidisciplinary Research and Development, 2(12), 1-9.
[57] Naser, S. (2009). Evaluating the effectiveness of the CPP-Tutor an intelligent tutoring system for students learning to program in C++. Journal of Applied Sciences Research, 5(1), 109-114.
[58] Shaath, M. Z., Al-Hanjouri, M., Abu Naser, S. S., & Aldahdooh, R. (2017). Photoshop (CS6) intelligent tutoring system. International Journal of Academic Research and Development, 2(1), 81-87.
[59] Hamed, M. A., & Abu Naser, S. S. (2017). An intelligent tutoring system for teaching the 7 characteristics for living things. International Journal of Advanced Research and Development, 2(1), 31-45.
[60] Al-Nakhal, M. A., & Abu Naser, S. S. (2017). Adaptive Intelligent Tutoring System for learning Computer Theory. EUROPEAN ACADEMIC RESEARCH, 6(10), 8770-8782.