Y. N. AMIRGALIYEV [1], A. KUTLU [2], A. V. BOGDANCHIKOV [1], K. N. LATUTA[1], R. N. SULIYEV [1]

([1]Suleyman Demirel University, Kaskelen, Republic of Kazakhstan)
([2]Suleyman Demirel University, Isparta, Turkey)

USING ANDROID

TO IMPLEMENT INTELLIGENT TESTING SYSTEM

Annotation. The first part of the paper discusses the role of mobile applications in person's life and proposes a method to apply testing system in Android device. The second part of the paper describes the algorithm used in the prototypes of the testing system. The third part explains details of each prototype and their key concepts. In conclusion the two prototypes are compared and advantages of each are highlighted.

Keywords: testing system, mobile devices, Android, adaptive algorithms.

Тірек сөздер: сынақтау жүйесі, мобильді құрылғылар, Android, адаптивті алгоритмдер.

Ключевые слова: система тестирования, мобильные устройста, Android, адаптивные алгоритмы.

Introduction. Progressive usage of smart-phones based on such platforms as Android, IOS, Windows Mobile give people opportunity to provide information and service via mobile technologies. Every person who has smart-phone tends to interact with it often. It is software developers' task to present people useful applications, such as connection to social media, which in turn can be used to improve lessons actuality and attractiveness [1].

Nowadays there exist a lot of tutoring systems that can be used to provide students with effective software, and developers tend to transform these systems to mobile platforms [2]. Using mobile device students can improve their knowledge everyday when they have spare time and desire. Mobile devices are compact and used in daily life therefore always carried along, so these devices are always on hand.

Whenever a student starts learning with the help of tutoring system, the progress and the speed of grasping process vary for each student. So it is important to provide different learning methods for each student. In this case some adaptive algorithms can be developed [3].

In this paper two different prototypes of single testing system are discussed. The first prototype is a simple list of tests, where user can select and answer each question in any order. In second prototype questions are divided by 10 levels of difficulty. When the student passes a small quiz of 10 questions the system increases difficulty level.

For the purpose of simplicity prototypes in this paper are based on questions about Python programming lan-guage, and both use database with 100 questions arranged in 10 complexity levels. However intelligent testing system can be used to work with materials of any course, such as Calculus, Digital Design or any other.

Figure 1 – Prototype 1, Main Window

Algorithm. Prototype 1 is very simple and initially supposed to present all testing material in raw format. So user can decide how to control learning process. Appli-cation of prototype 1 contains list of 100 questions and user can select one of them. Once the question is selected a window with question text appears, and the user can give answer. After user answers status of that question changes to correct or incorrect according to result, see figure 1. The user also can reset his progress anytime to start from the beginning.

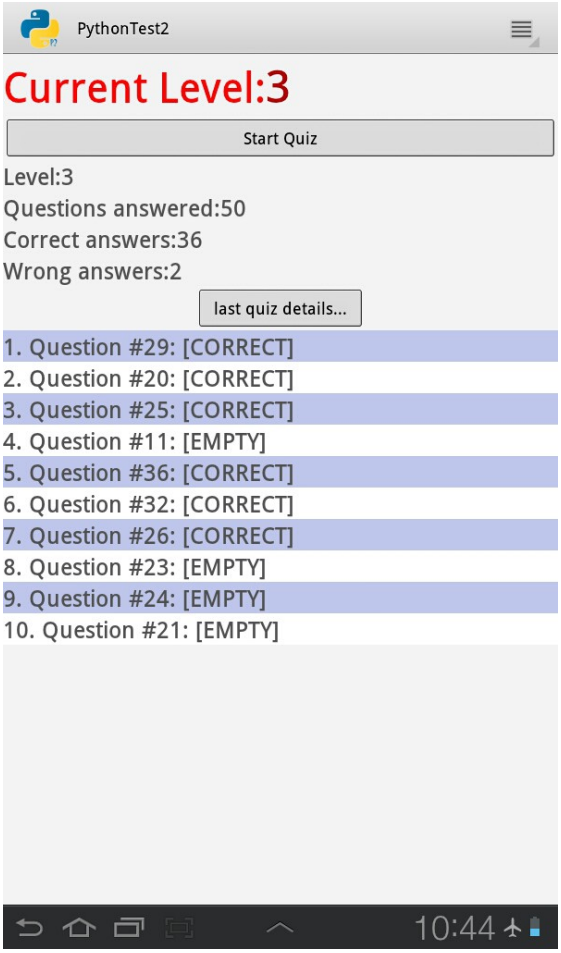
Algorithm of Prototype 2 is a bit more complex; it includes 10 levels of difficulty and a user starts from level 1. If user starts the quiz and scores more than 8 out of 10 the system increases the difficulty level. But if user answers correctly to less than 3 questions system will decrease the level, otherwise difficulty level remains the same.

To make testing process more effective, questions in a quiz are selected in random manner and difficulty depends on current level. Randomness of quiz questions is decided by Gaussian distribution, so some times questions from lower or higher level can appear, and that makes the challenge more exciting. The results of each quiz are sent to evaluation module, where correct, wrong and empty answers are counted. If the number of correct answers is greater than 8, module switches user's current level to next one, if possible. But if the number of correct answers is less than 3, it decreases current level, if possible, otherwise does nothing.

Implementation. Both prototypes are implemented on Android platform using native programming language JAVA. Implementation of Prototype 1 is straight forward; it displays one ListActivity with list of questions and a small icon for each question where icon type defines the status, like it is shown in Figure 1. Activity is opened if user taps on one of listed items. This Activity loads question text from SQLite database, which stores all questions and answers in separate tables. In Activity user reads question and fills the answer into corresponding EditText field, and tap "answer" button. After that, result is compared with correct value from answers table in database. If it is correct, question status changes to OK icon. If it is wrong, question status changes to WRONG icon. For any questions that are not answered yet icon remains as QUESTION icon.

Menu in initial ListActivity shows three items: *statistics* item which shows statistics of answers for the ques-tions, *clear* item that is used to reset status for all questions, and *about* item that shows authors of this application.

Realization of Prototype 2 presents more complex structure. It contains MainActivity, where useful information about user's progress is shown. Whenever user presses on "Start Quiz" button, new Quiz Activity is created which in turn requests the LevelController to generate new Quiz from data stored in SQLite database.



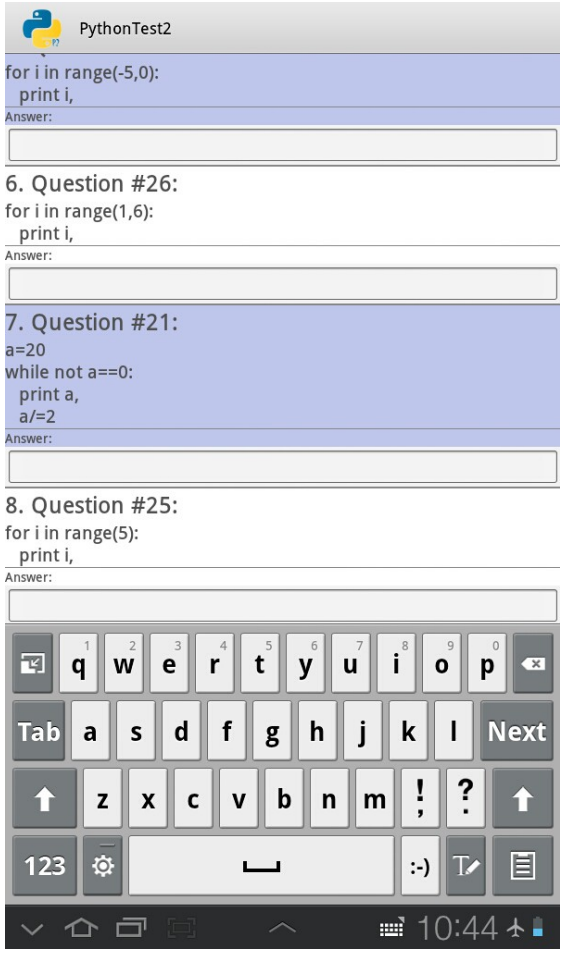


Figure 2 – Prototype 2, MainActivity

Figure 3 – Prototype 2, QuizActivity

This application consist of three logical elements: MainActivity – to display current level, statistics and results of last quiz, see figure 2; QuizActivity – to show generated quiz questions and receive answers for each question, as shown in figure 3; LevelController – to access to database, generate quizzes, and make important decisions to change level value.

One of the important parts is implementation of LevelController's quiz generation process and making decision of level progress. Generating quiz is implemented using Gaussian distribution, with mean (μ) = current level and variance (σ^2) = 0.4.

So the resulting distribution of questions will be grouped close to current difficulty level and sometimes will produce one or two questions from adjacent levels. Figure 4 shows part of Java code that uses Gaussian distribution to generate new quiz.

```
for(int i=0;i<period;i++){
```

```
        int lvl = (int)Math.round(curLevel+r.nextGaussian()*0.4);
        if(lvl<1 || lvl>10){i--;continue;}
        boolean ok = false;
        do{
                int p = r.nextInt(period)+period*(lvl-1);
                if(!taken.contains(p)){
                        taken.add(p);
                        q.addQuestion(p, questions.get(p));
                        ok = true;
                }
         }while(!ok);
}
```

Figure 4 – Prototype 2, Code to generate new quiz

Decision of level changing is done based on result of answers check. All given answers are compared with answers in database and number of equal values is counted. Then this value is compared with limits according to which current level is increased or decreased. Implementation of this process is shown in Figure 5.

```
int correct = 0, wrong=0;
for(int i=0;i<myAns.length;i++){
        if( myAns[i].equals(answers.get(q.getId(i))) ){
                correct ++;
                edit.putString("q"+i, (i+1)+". Question #"+q.getId(i)+": [CORRECT]");
        } else if(myAns[i].length()>0){
                 wrong ++;
                edit.putString("q"+i, (i+1)+". Question #"+q.getId(i)+": [WRONG]");
        } else
```

```
                edit.putString("q"+i, (i+1)+". Question #"+q.getId(i)+": [EMPTY]");
}
if(correct<3 && curLevel>1) curLevel--;
else if (correct>8 && curLevel<10) curLevel++;
```

Figure 5 – Prototype 2, decision making code

Conclusion. The use of mobile devices can efficiently improve educational process. Students and instructors can benefit from mobile applications. As described in this paper, Android application for testing student knowledge can be easy way to motivate students to learn by answering questions on personal Android based smart-phone.

The prototype 1 application is good way to provide students with all materials and to check know-ledge. It has concise and user-friendly interface, understandable navigation system, and internal database of questions for the course. Disadvantage of this prototype is that it becomes boring and very annoying to answer every question one by one, without level gradation that helps to evaluate knowledge.

Prototype 2 turns testing program into intellectual challenge for students. It keeps track of all attempts and shows correct and wrong answers. Moreover it chooses quizzes that are most appropriate for current level of competence in this course. Random distribution of questions among quizzes forces student to repeat some already answered questions and remember their answers better. Disadvantage of this scenario is that students who get stuck in one level will never see questions from more complex topics. Also the process of preparing questions and answers database requires a lot of work from teacher of that course.

Later these prototypes can be improved to make adaptation algorithm which will work without any difficulty limits and which will fit the user's knowledge. Also it is possible to insert different hint materials to help students overcome most challenging questions of the course.

In conclusion, combining mobile technologies with educational process can boost students' interest and learning speed, so that students can efficiently and continuously progress in the study of a course.

REFERENCES

1 Aitchanov B.H., Satabaldiyev A.B., Bogdanchikov A.V., Latuta K.N. Innovative approach to social media in education // News of the National Academy of Sciences of the Republic of Kazakhstan, Series of Physics and Mathematics #4(290) 2013.

2 Vlado Glavinić, Marko Rosić, Marija Zelić. Extending Intelligent Tutoring Systems to Mobile Devices. Springer, 2008.

3 Annabel Latham, Keeley Crocket, David McLean, Bruce Edmunds. A conversational intelligent tutoring system to automatically predict learning styles. Elsevier, 2012.

4 Reto Meier. ”PROFESSIONAL ANDROID™ 2 APPLICATION DEVELOPMENT”, Wiley Publishing, Inc. ISBN 978-0-470-56552-0, 2010.

5 Allen Downey, Jeffrey Elkner, Chris Meyers. How to think like a computer scientist: learning with Python. Green Tea Press. ISBN 0-9716775-0-6, 2008.

Резюме

Е. Н. Амиргалиев[1], А. Кутлу[2], А. В. Богданчиков[1], К. Н. Латута[1], Р. Н. Сулиев[1]

([1]Университет им. Сулеймана Демиреля, Каскелен, Республика Казахстан,
[2]Университет им. Сулеймана Демиреля, Испарта, Турция)

ИСПОЛЬЗОВАНИЕ ANDROID ДЛЯ РЕАЛИЗАЦИИ ТЕСТИРУЮЩЕЙ СИСТЕМЫ

В статье описываются два прототипа интеллектуальной тестирующей системы. В первой части этой статьи описывается роль мобильных приложений в жизни человека и предлагается метод для использования тестирующей системы на устройствах Android. Во второй части описаны алгоритмы, которые используются в каждом из прототипов. В третьей части объясняются детали каждого из прототипов и основные концепции их реализации. В конце два прототипа сравниваются и выделяются преимущества каждого.

Ключевые слова: система тестирования, мобильные устройста, Android, адаптивные алгоритмы.

Резюме

Е. Н. Әмірғалиев[1], А. Кутлу[2], А. В. Богданчиков[1], К. Н. Латута[1], Р. Н. Сулиев[1]

([1]Сүлейман Демирел атындағы университет, Қаскелең, Қазақстан Республикасы,

[2]Сүлейман Демирел атындағы университет, Испарта, Түркия)

ANDROID-ты СЫНАҚТАУ ЖҮЙЕЛЕРІНДЕ ҚОЛДАНУ

Мақалада зерделі сынақтау жүйесінің екі прототипі сипатталады. Мақаланың басында мобильді қосым-шалардың адам өміріндегі орны және олардың Android құрылғыларында қолданысы келтірілген. Мақаланың екінші бөлімінде ұсынылған прототиптерді құру қағидалары мен іске асыру алгоритмдері тиянақты сипат-талады. Мақала соңында ұсынылған прототиптердің артықшылықтары қарастырылып, олардың өзара салыс-тыруы орындалған.

Тірек сөздер: сынақтау жүйесі, мобильді құрылғылар, Android, адаптивті алгоритмдер.

Поступила 04.06.2013 г.