

## DEVELOPMENT OF WEB - BASED MOBILE SYSTEM FOR TRAINING AND SELF-CONTROL

Dimitar Pilev, *University of Chemical Technology and Metallurgy, d\_pilev@yahoo.com*

**Abstract:** Development of mobile communication technology sets new standards for elearning and distance learning platforms. The use of smartphones and tablets increasingly entering into the lives of consumers. The need for constant access to the resources regardless of time and place requires not only the development of standard web-based applications, but also applications for mobile devices. This article is devoted to topical issues related to the design and development of Web-based mobile system for distance learning and self-control.

**Key words:** e-learning, m-learning, mobile applications, distance learning.

## РАЗРАБОТВАНЕ НА МОБИЛНА WEB – БАЗИРАНА СИСТЕМА ЗА ОБУЧЕНИЕ И САМОКОНТРОЛ

Димитър Пилев, *Химикотехнологичен и металургичен университет, d\_pile@yahoo.com*

**Абстракт:** Развитието на мобилните и комуникационни технологии налага нови стандарти за разработване на платформи за електронно и дистанционно обучение. Използването на смартфоните и таблетите навлиза все-повече в живота на потребителите. Необходимостта от постоянен достъп до предоставените ресурси, независимо от времето и мястото, налага разработването не само на стандартни Web-базирани приложения, но и на приложения за мобилни устройства. Настоящата статия е посветена на актуалния проблем, свързан с проектиране и разработване на мобилна Web–базирана система за дистанционно обучение и самоконтрол.

**Ключови думи:** e-learning, m-learning, дистанционно обучение, мобилно приложение

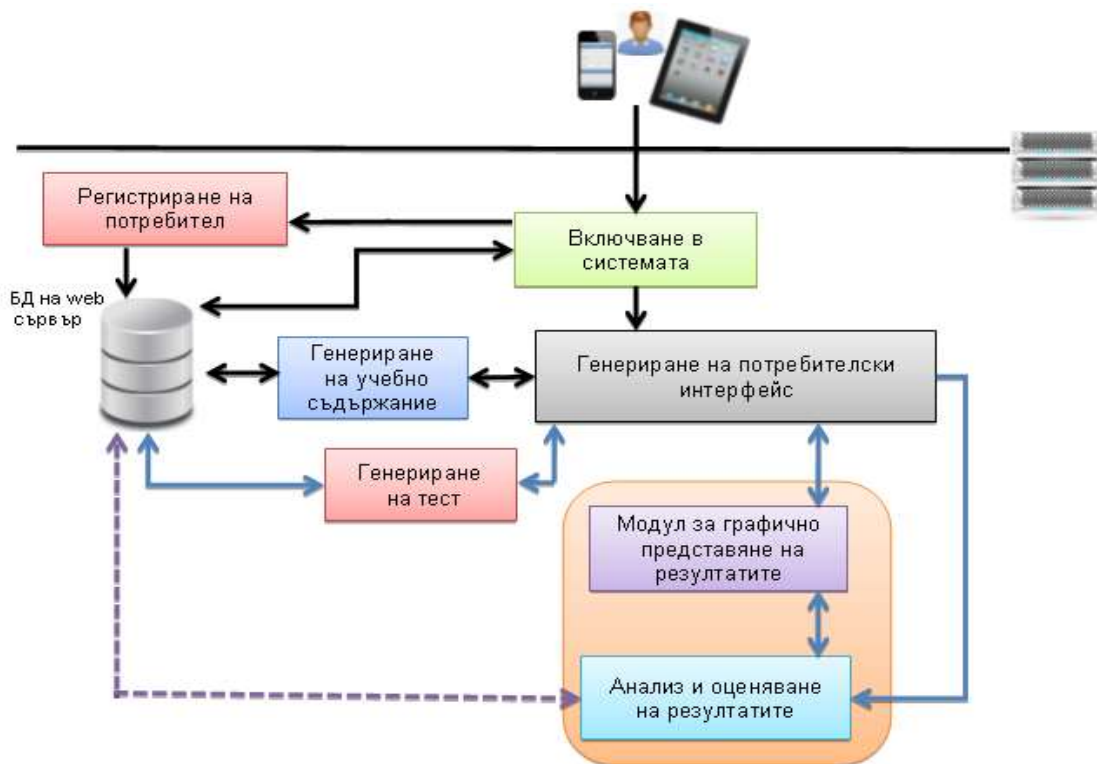
### Въведение

Развитието на мобилните и комуникационни технологии налага нови стандарти за разработване на платформи за електронно и дистанционно обучение [1], [2], [3]. Наличието на интернет позволява разработването и използването на Web–базирани информационни системи. Основните предимства на съвременните Web – базирани системи са независимост от използваната платформа, възможността за едновременно използване от много потребители по всяко време и наличието на универсално клиентско приложение. Внедряването на такива системи повишава качеството на електронното и дистанционно обучение [4].

С развитието на технологиите смартфоните и таблетите станаха неразделна част от живота на потребителите. Необходимостта от постоянен достъп до предоставените ресурси,

независимо от времето и мястото, налага разработването не само на стандартни Web-базирани приложения, но и на приложения за мобилни устройства.

Съвременните мобилните приложения са приложения, разработени за малки преносими устройства като мобилни телефони, смартфони, планшети и др. Те са базирани върху отворени стандарти (HTML, CSS и JavaScript) и не отстъпват по възможности на традиционните десктоп приложения с графичен потребителски интерфейс. Като се отчита популярността на мобилните устройства за работа и комуникация, разработването и реализацията на мобилни приложения се явява изключително актуална тема. Ефективната разработка на мобилни приложения стана възможна благодарение на използването на стандартизирани JavaScript библиотеки (jQuery Mobile и др.) [5], [7]. Сред основните функции, предоставяни от тези библиотеки, са независимият от браузъра асинхронен достъп до сървърни ресурси, добил популярност под името AJAX [7], както и наличието на множество готови компоненти за реализиране на анимирани ефекти и др.



Фиг. 1 Архитектура на мобилна Web-базирана система за обучение и самоконтрол

### Архитектура на мобилното WEB – приложение

На фиг.1 е представена архитектурата на мобилната система за обучение и самоконтрол. Системата включва следните модули:

- Модул за включване в системата

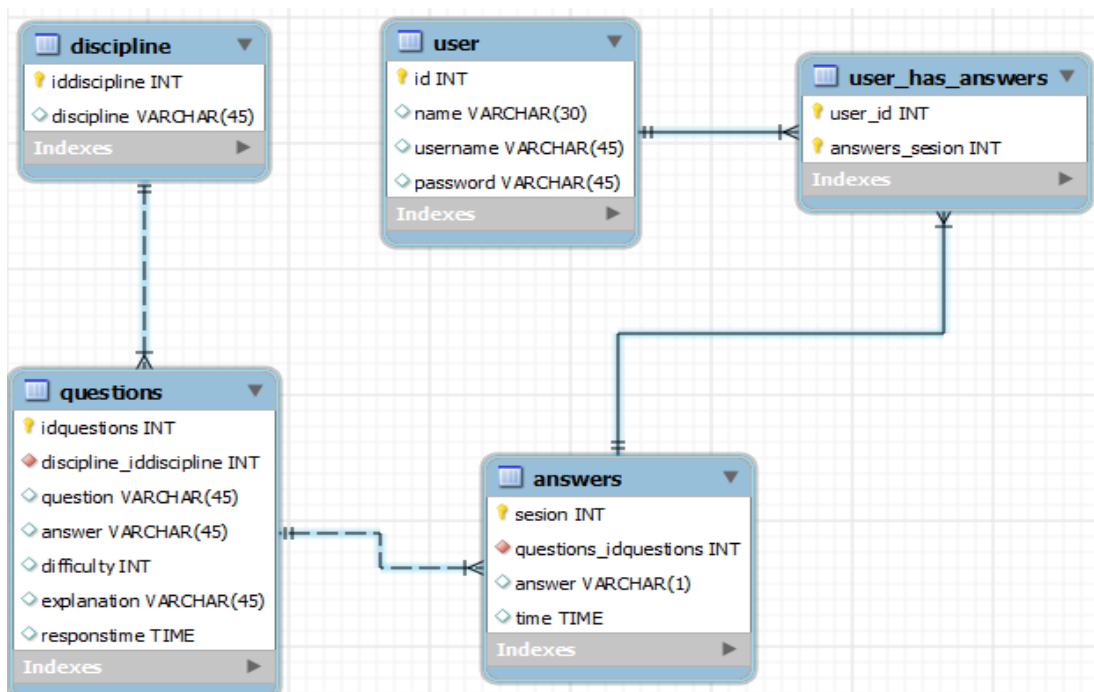
Тук се извършва проверка за налична регистрация. Включването в системата се осъществява чрез потребителско име и парола.

- Модул за регистриране в системата

За достъп до ресурсите на мобилната система е необходима регистрацията на потребителя.

- Модул за генериране на учебно съдържание  
В този модул по случаен начин се генерират въпросите и верните отговори, както и свързаните с материала пояснения. Използва се база от данни. Въпросите се генерират в зависимост от избраната от потребителя дисциплина и ново на трудност. Броят на генерираните въпроси се задава от потребителя.
- Модул за генериране на потребителски интерфейс в зависимост от избрания режим за обучение или самоконтрол.
- Модул генериране на тест  
Генерира тестови въпроси в режим на самоконтрол. Характерно за модула е, че за всеки въпрос се задава и времето му за отговор.
- Модул за анализ и оценяване на резултатите  
Обработва и анализира резултатите от проведения тест. За целта се извлича информация от базата от данни. На базата на резултатите от тестовете и придвижването на обучавания по дървовидната структура на знанието са използвани механизми за извличане на историята на обучението, модела на обучаемия и непълнотите на неговите знания[8].
- Модул за графично представяне на резултатите  
Визуализира в графичен вид резултатите от проведения тест.
- База от данни на web-сървър

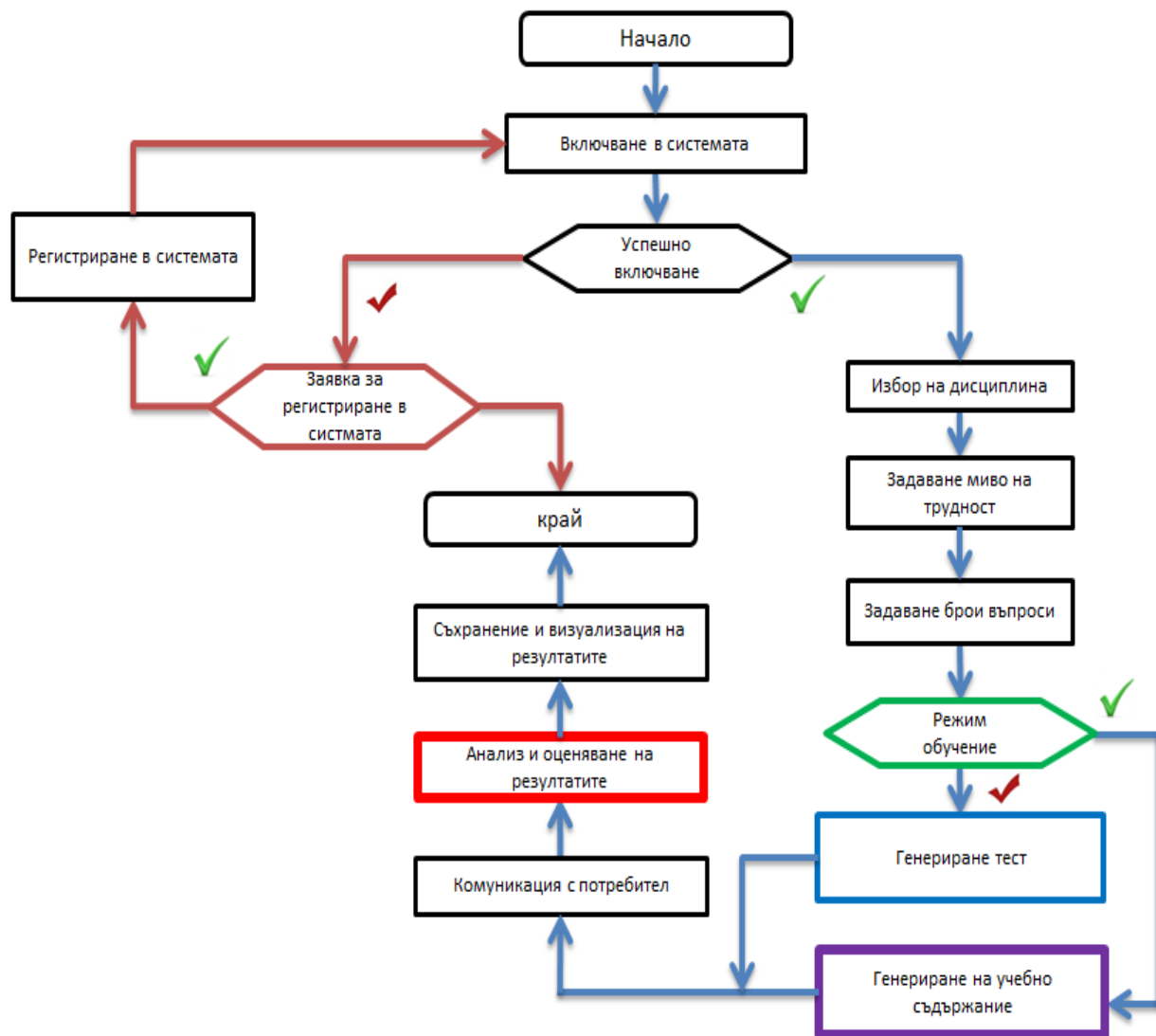
На Фиг.2 е представена схема на реляционна база от данни, използвана в системата за обучение и самоконтрол.



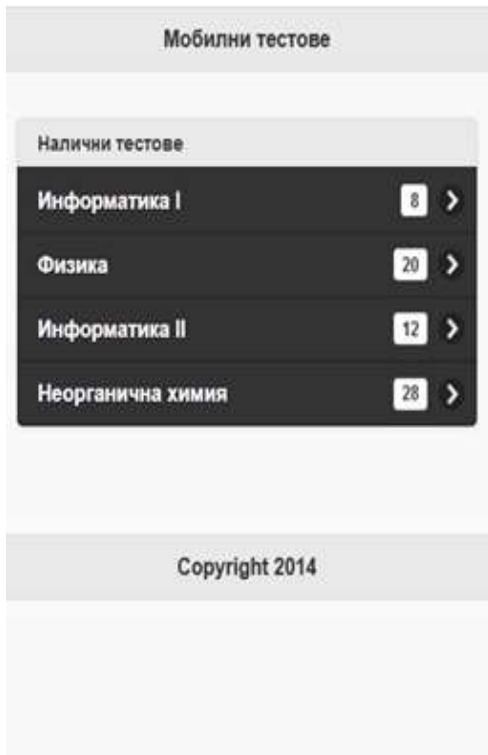
Фиг. 2 Схема на реляционна база от данни, използвана в системата за обучение и самоконтрол

### Реализация на системата

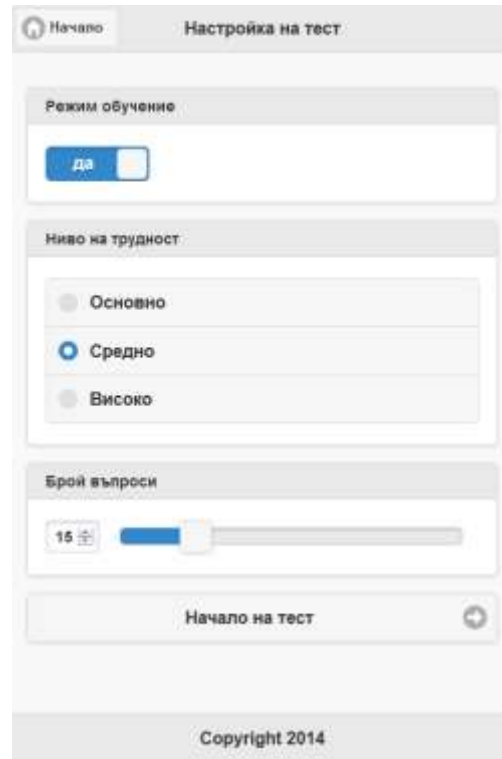
На Фиг. 3 е представен алгоритъмът, по който функционира мобилната система за обучение и самоконтрол. Потребителят избира дисциплината и нивото на трудност, брой на въпросите. В режим на обучение системата генерира тестови въпроси, съпроводени с верни отговори и пояснения. Верните отговори, както и поясненията за отделните въпроси се предоставят при поискване от потребителя. Тази допълнителна функционалност се реализира от страна на клиентското приложение. Характерно за този режим е, че не се задава време за отговор на отделните въпроси. При тестване знанията на потребителя, системата генерира единствено задаваните въпроси от избраната дисциплина и ниво на трудност. За всеки отделен въпрос се задава и време за отговор, което е съобразено със сложността на въпроса. По този начин не се допуска използване на допълнителни източници за информация от обучаемия.



Фиг. 3 Алгоритъм, използван от системата за обучение и самоконтрол



Фиг. 4 Начална страница на системата



Фиг. 5 Настройка на режима на обучение или самоконтрол

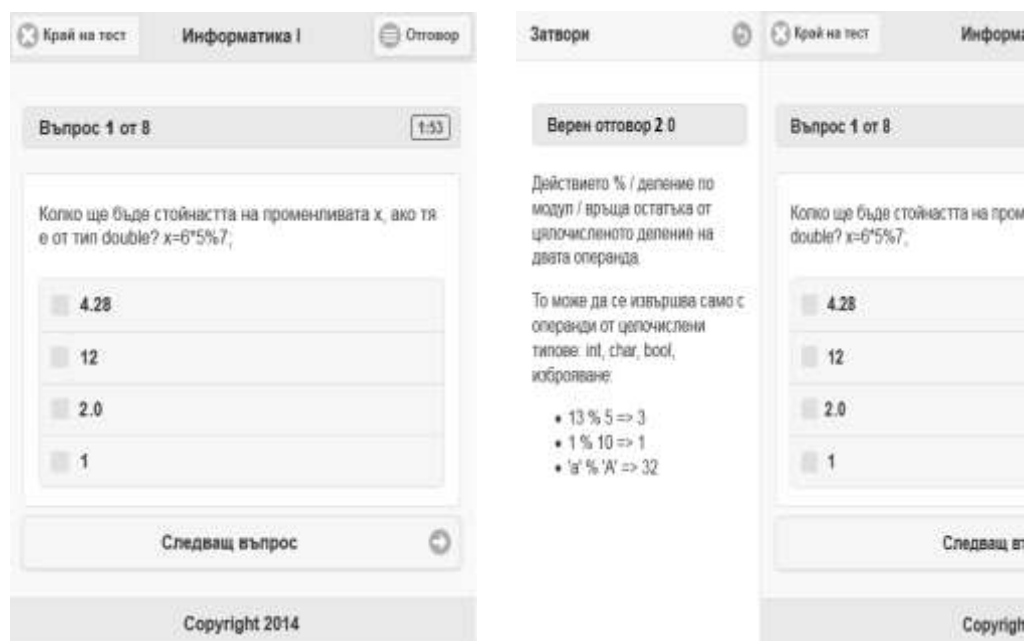
При успешно включване потребителят се пренасочва към модула, генериращ потребителски интерфейс. Първоначално в този модул се генерира началната страница на системата, в която са показани дисциплините, по които може да бъде провеждано обучението и/или тестването на знанията (фиг.4). След избор на дисциплина се задава ниво на трудност, броят на тестовите въпроси и режим на обучение или самоконтрол (фиг.5).

В режим на обучение генерираните от модула „Генериране на учебно съдържание” въпроси се предават към модула „Генериране на потребителски интерфейс” за представяне по подходящ начин. Въпросите, отговорите, както и свързаните с тях обяснения се извличат от база от данни. Първоначално се визуализира само частта от Web-страницата, показваща въпроса на потребителя. При поискване се показва и верният отговор, съпроводен с подходящи пояснения (фиг.6). Тази функционалност се осъществява от страна на клиентското приложение, като за целта се използва JavaScript базираната библиотека jQuery Mobile. По този начин не се изпращат допълнителни заявки към web-сървъра, което пести трафик и процесорно време на сървъра.

Резултатите от проведените тестове се съхраняват в база от данни. Това дава възможност за диференциране на обучаваните чрез клъстерен анализ [9]. Всеки регистриран в системата потребител може да проследи своето развитие в процеса на обучение.

Генерирането на тестови въпроси в режим на самоконтрол се осъществява от модула „Генериране на тест”. Въпросите от теста се подават отново към модула „Генериране на потребителски интерфейс”. При изтичане на времето за отговор, данните от формата, съдържаща отговорите на потребителя, автоматично се изпращат към Web-сървъра, след което се показва следващият въпрос. За тази цел се използват Ajax заявки.

Основната функционалност на потребителския интерфейс в режим на сомоконтрол се осъществява от функция *getQuestion* (фиг. 7), реализираща комуникацията между клиента и Web-сървъра. Функцията се изпълнява от страна на клиента, като основните ѝ дейности са свързани с изпращане на отговорите на зададените въпроси обратно към сървъра (модул „Анализ и оценяване на резултатите“) и визуализиране на генерираните от системата въпроси. Самата функция може да бъде извикана по два начина: чрез натискане на бутона „следващ въпрос“, изпращащ данните от формата, съдържаща изборния от потребителя отговор (редове 40 – 46 от Фиг.7) или изтичане на времето за отговор на указания въпрос, зададено чрез променливата *answerTime*. Последната функционалност се реализира чрез таймер (ред 34 от Фиг.7), извикващ повторно функцията *getQuestion* след изтичане на времето за отговор на зададения въпрос. Таймерът се анулира при натискане на *submit* бутона, изпращащ резултатите към сървъра, от страна на потребителя (ред 41 Фиг.7).



Фиг. 6 Потребителски интерфейс в режим на обучение

Специфично за разработената система е начинът, по който се обработват грешки, възникнали поради забавянето на отговора на клиентската заявка от страна на Web-сървъра или липсата на такъв. Това важи в пълна сила при мобилните приложения, където скоростта на интернет връзката зависи от обхвата на мобилната мрежа. Забавянето или липсата на отговор от страна на Web-сървъра предизвиква грешка в браузера на клиента. За да се осъществи отново достъп до системата, потребителят трябва да актуализира страницата за достъп до приложението. При ръчно актуализиране на Web-страницата, системата ще генерира грешка, породена от десинхронизацията между клиента и Web-сървъра. Честото възникване на грешки от този тип обаче, води до неефективното използване на web-базираната система. За отстраняване на този недостатък се задава времето за отговор на Web-сървъра (ред 2, Фиг.7). При липса на отговор, в указаното време, се изпраща повторна заявка към сървъра. При наличие на *n* на брой неуспешни опити за връзка със сървъра (редове 15 и 16 от Фиг. 7), провежданият тест се прекратява и

потребителят се пренасочва към страница, извеждаща съобщение за грешка. Времето за отговор на сървъра, както и броят на неуспешните опити за връзка се задават в зависимост от типа на провеждания тест.

```

1   var countErr = 0,
2       responseTime = 5000,
3       answerTime = 75000,
4       unsuccessfulAttempts = 3,
5       timer;
6   function getQuestion() {
7       $.ajax({
8           url: 'question.php',
9           type: 'POST',
10          data: $('#result').serialize(),
11          timeout: responseTime,
12          dataType: 'json',
13          error : function () {
14              countErr++;
15              if(countErr >= unsuccessfulAttempts) {
16                  $.mobile.changePage('error.php?err=1');
17              }
18          },
19          success: function(data) {
20              if(data.hasOwnProperty('err')) {
21                  $.mobile.changePage('error.php?err=' + data.err);
22              } else if(data.qNum=="end") {
23                  $.mobile.changePage('finish.php');
24              }
25              else {
26                  countErr = 0;
27                  /*
28                   код за визуализация на въпроса
29                  */
30                  answerTime = data.responsetime;
31              }
32          },
33          complete: function() {
34              timer = setTimeout(getQuestion, answerTime);
35          },
36          cache: false
37      });
38  }
39
40  $('#result').submit(function(event) {
41      clearTimeout(timer);
42      event.preventDefault();
43      getQuestion();
44  });

```

Фиг. 7. Програмна реализация на функция `getQuestion`, осъществяваща комуникацията между клиента и web-сървъра в режим на самоконтрол

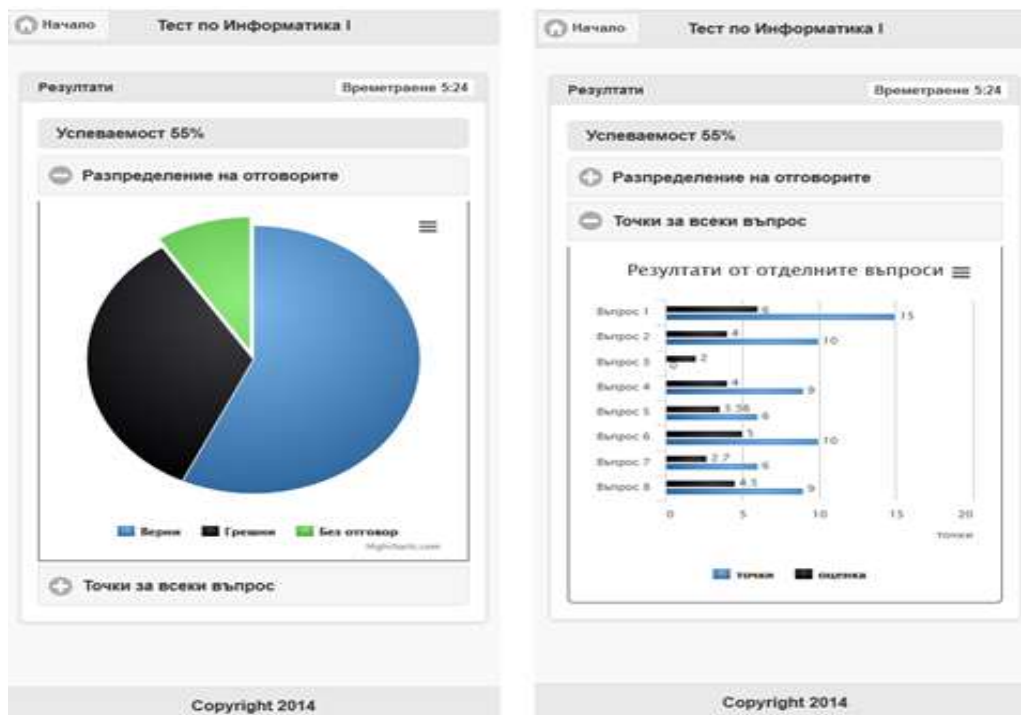
Генерираните от модула „Генериране на тест” въпроси се изпращат към клиента в JSON [10] формат (Фиг. 8).

```
$questionInfo = [  
  "qIDs" => "45",  
  "qNum" => "2",  
  "question" => "Колко ще бъде стойността на променливата x, ако тя е от тип double? x=6*5%7;",  
  "a" => "4.28",  
  "b" => "12",  
  "c" => "2.0",  
  "d" => "1",  
  "responsetime" => "115000"  
];
```

Фиг. 8. Формат на въпросите, изпращани от системата в режим на самоконтрол

Изпращаните въпроси, съдържат данни за:

- qIDs – сесиен идентификационен номер на въпроса;
- qNum – пореден номер на въпроса от провеждания тест;
- question – генерираният от системата въпрос;
- възможните отговори - a,b,c,d;
- времето за отговор на въпроса, зададено в милисекунди.



Фиг. 9. Визуализация на резултатите от проведен тест

При изчерпване количеството на тестовите въпроси, като стойност на атрибута qNum се задава „end”. По този начин потребителят се пренасочва към страница, обозначаваща приключването на провеждания тест (редове 22 и 23 от Фиг.7).

В случай, че в системата възникне грешка, изпращаните от Web-сървъра данни имат следния формат:  $\$questionInfo = ["err" => "errorCode" ]$ . Чрез стойността на атрибута err, потребителят се уведомява за възникналата грешка (редове 20 и 21 от Фиг.7).

Визуализацията на резултатите, генерирани от модула за графично представяне на резултатите е представена на Фиг. 9.

## **Заклучение**

Разработена е мобилна Web – базирана система за дистанционно обучение и самоконтрол по различни дисциплини. Системата може да се използва както при наличие на смартфон и таблет, така и през стандартен web-браузър. Използвани са отворени стандарти, което прави системата платформено независима. Функционалността на системата е разпределена между клиентското приложение и Web – сървъра. Използването на AJAX заявки намалява трафика. По този начин се пестят време и ресурси в процеса на обучението.

## **Литература**

- [1] Som Naidu, A Guide book of Principles, Procedures and Practices, Revised Edition, CEMCA, 2006, ISBN: 81-88770-04-3
- [2] A. Georgieva, D. Pilev, Web-based Programming Course – Concepts and Performance, Journal of the UCTM, volume 41, ISSUE 2, 2006, ISSN 1311-762
- [3] Mohamed Ally, Avgoustos Tsinakos, Increasing Access through Mobile Learning, Commonwealth of Learning and Athabasca University, 2014
- [4] Халачев П., Й. Ангелова, Компютърните тестове – метод за контрол на знанията, Сборник доклади от Юбилейна научна конференция – 50 години катедра Икономика и стопанско управление“, ХТМУ, 03.06.2013 г
- [5] jQuery Mobile, A Touch-Optimized Web Framework, Available from: <http://jquerymobile.com>, visited 2014
- [6] Creating Web Apps with Dojo, Available from: <http://dojotoolkit.org>, visited 2014
- [7] w3schools, AJAX, Available from: <http://www.w3schools.com/ajax/default.asp>
- [8] V. Nikolov, St. Panov, Interaction between subject area model, student model and planning activities in learning process, Computer & Communications Engineering, Vol 7' No 2, 2013
- [9] Халачев П., Модел за диференциране на обучаваните чрез клъстерен анализ, Международна научна конференция УНИТЕХ 13, ТУ Габрово, 22-23.11.2013 г., Габрово, стр. 397-401
- [10] The JSON Data Interchange Format, Standart ECMA-404, 1st Edition / October 2013, available: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>