

**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ СОФИЯ**

**КУРСОВ ПРОЕКТ**

***ПО***

***Паралелно Програмиране***

***Тема:***

***Мобилно приложение***

**ИЗРАБОТИЛ:**

*Хрисимир Емилов Даков*

Факултетен номер:

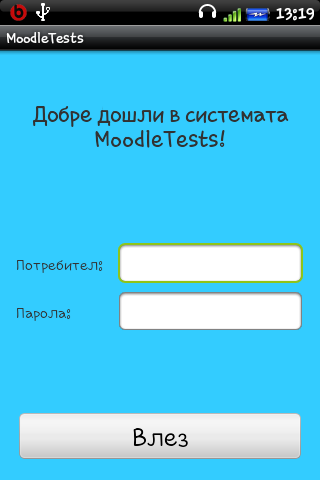
121210019

Описание:

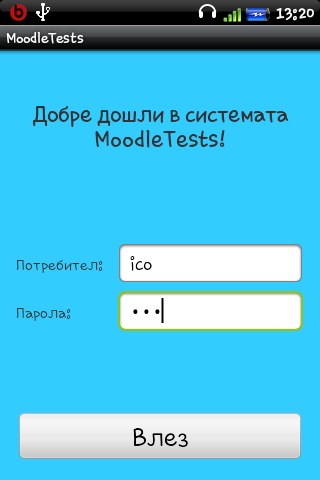
Приложението предоставя възможност за правене на онлайн тестове от мобилни устройства с Android.

Има следната функционалност:

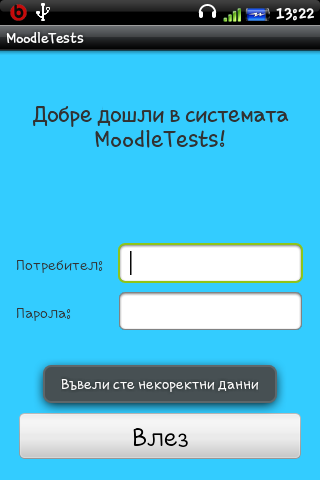
1.При влизане в приложението се показва логин екран в който да се въведат потребителско име и парола на студента.



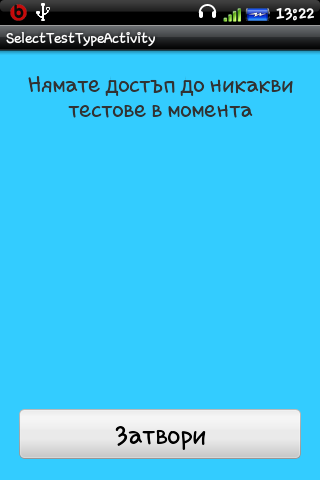
2.При попълване на правилни данни и „Влез”



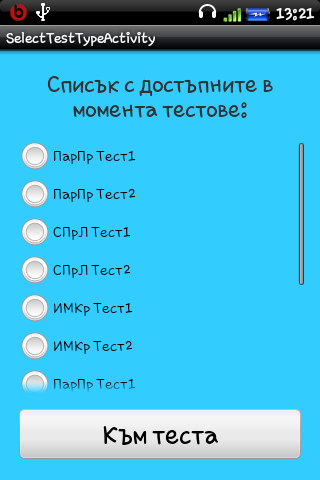
3.Прави заявка към сървъра за да види дали данните имат достъп до налични тестове ако не сме въвели нищо:



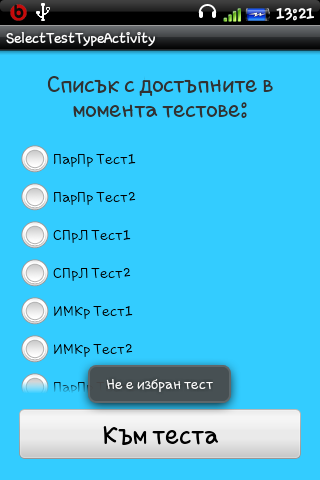
Или данните за потребителя не са открити:



При успешно влизане взима списък с наличните тестове които имаме право да се опитаме да решим от сървъра :



Ако не сме избрали и цъкнем „Към теста”:

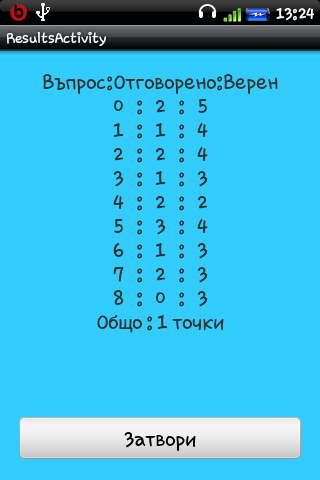


При продължаване напред се прави заявка към сървара да ни даде списък с въпросите от теста. Вслучая сървърът е реализиран като клас със статични методи , кото генерират случайни по брой и дължина въпроси и отговори от които ние да избираме според нас правилният и да продължим към следващия въпрос до края на списъка. Бутоните за назад и меню са деактивирани за да се избягва евентуална поправка в следствие на преписване.



По време на решаване на теста отговорите се пазят в динамичен масив от интове с големина като на списъка с въпросите с цел по лесна обработка и изпращане между активититата.

След като сме завършили теста се изкарва статистика за това как е протекъл теста:



Поради проблеми с Moodle сървърът е реализиран като клас със статични методи , но структурата на приложението и организацията на кода дават възможност за взимане на данните чрез връзки със Java приложения (Sockets , HttpRequest ) и дори може да се развие в насока да взима тестовете от RSS feed чрез помоща на SAX парсър за по голяма производителност. За прилагане на горните методи е необходимо само да се промени имплементацията сървърния клас а останалата част от приложението ще остане непокътната.

Приложението се състои от следните класове:

1.StartActivity

–това е активитито което се пуска при стартиране на приложението и което изкарва логин екрана.

Съдържа функция login() чрез която се създава интент(

Това е начина за създаване на нов прозорец в андроид:

Intent i = new Intent(this,SelectTestTypeActivity.class);

В конструктура може да бъде подаден и стринг , който да съвпада с интент филтъра на други активитита , и така може да се пусне компонент който дори не е на нашето приложение.

)

, добавят се потребителските данни като екстри и се извиква следващото активити.

(

i.putExtra("usr", user);

i.putExtra("pwd", pwd);

startActivity(i);

Екстрите са допълнителни данни които се прикачат към интенда, който пуснатото със него активити може да извлече чрез getIntent().getExtras(), и чрез тях се предава информация между активититата.)

2.SelectTestTypeActivity –зарежда списъка с възможните тестове които можем да решим след като сме ги взели от сървъра с addTestAvailable() , aко сървъра не ни е върнал списък , значи логина е грешен и се изкарва съобщение за това. Бутона се променя на „Затвори”.

При избор на тест и потвърждаване се пуска следващтото активити на което като екстра се предава ID на теста , за да може да си вземе въпросите от сървъра.

3.MakeTestActivity-активитито взима въпросите с getQuestions() ,след което се вика startTest() ,тя от своя страна вика nextQuestion(vaprosId) ,която се вика рекурсивно докато не свършат въпросите. Накрая масивът с отговорите се праща като екстра към последното активити - 4.ResultActivity където се прави заявка за взимане на правилните отговори и се извежда статистика спрямо дадените за целия тест.

5.MoodleServer –това е класът към който се обръщаме за тестове а после и въпроси . Методите му само генерират случайни данни ,но при различна тяхна имплементация би работило без проблеми.

6.Question – помощен клас който описва един въпрос – съдържа поле uslovie , и списък vazmojni\_otgovori , и спомага за комуникацията в приложението.

Освен тези класове са използвани и андроидските ресурс файлове „strings.xml” където са зададени всички текстове на приложението за евентуален лесен преход към друг език . Целият UI на приложението е ръчно описан в XML layouts без употребата на графични редактори които понякога предизвикват нежелано поведение.Вмъкнати са коментари по сорскода .Навсякъде се правят проверки за верността на данните .

Прилагам архив с проекта.

Благодаря за отделеното време!