мисля, че възможно най-близко се опитах да се доближа до архитектурата за документацията, от тук надолу мисля, че няма какво да се вземе.

## Потребители които ще го ползват и Характеристики

Този продукт e предназначен за използване от студентите на Пловдивският Университет „Паисий Хилендарски“.

Продуктът е изцяло разработен за да дава възможността на студентите да си проверят своите знания по избраните от тях дисциплини, както по време на изпит така и извън него, просто за проверка на достигането моментно знание и ниво.

Освен това приложението може и да се използва от :

деца от Началното училище

ученици от Средното училище

ученици от Основно училище

С лека корекция на данните и обновяване на базата данни тази апликация може да се превърне в забавна и интересна игра дори и за по – малки деца в Детската градина.

Освен това може да се използва от преподаватели за да се извършва форма на контрол, за различна дисциплина, която са изучавали в дадения триместър.

*От гледна точка на характеристиките за продукта тук ще споменем само най – важните, които позволяват на продукта да работи и на потребителят да го използва пълноценно.*

Те са :

Свързване на база данни

Четене от база данни

Генериране на произволни въпроси от базата данни

Решаване на теста

Точкуване на теста

Записване крайната оценка и друга информация в база данни

## Работна среда

Софтуерът е писан специално за телефони,които работят под операционна система Android. Ползван е Android, защото до ден днешен на пазара това е най-известната и ползвана операционна система за телефон. Освен това позволява да се разработват приложения за нея и е напълно безплатна, раздвижена и лесна за използване операционна система.

Заeдно с Андроид операционната система идва вградена база данни която е използвана за приложението. Тя е SQLite база данни, която сама по себе си е лека и удобна за използване, но не е пригодена за тежки приложения и въвеждане на огромна информация в нея.

Освен SQLite IDE-то за писане на Андроид приложения идва със стандартен вграден емулатор, който не се препоръчва за използване поради бавната му работа, трудното и почти невъзможно управление. За целта ( тестване ) може да се ползва плъгина Genymotion, който дава доста по-бързи емулаторни устройства или просто ако имате телефони с андройд операционна система да го свържете с IDE чрез ADB Drivers и да работите директно с него.

## Ограничения за Дизайна и Имплементацията

По принцип за приложението не би трябвало да съществуват някакви ограничения, но това не е съвсем вярно, тъй като за да проработи то трябва да се подкарва на телефони с по - нова версия на Android.

Тя трябва да е поне над API LEVEL 19 или така наречената версия “Kit Kat” за Android.

Препоръчително е в нашата ситуация да се използва API LEVEL 20 и нагоре за да няма проблеми с новите възможности, който са добавени в Marshmellow и нагорни версии.

**2.7 С какво разполага системата ?**

За приложението в бъдеще се очаква да се разработи една ранкинг система, която ще поставя на почетно (отличително) място тези студенти с най - добри точки. Като съответно те ще получават и бонуси за доброто си представяне с теста който са решили и това съответно ще им помогне да се отличат от останалите студенти, като ще направят добро впечатления на своите преподаватели.

# Външни интерфейсни изисквания

## Потребителски интерфейс

Той се състои от :

* Edit text - ове ( widget ) за полета, които се попълват от студентите
* Text View – та (widget) за въпросите
* Кръгъл Progress Bar, за да следи за оставащото време
* Използвани са също custom TOAST съобщения за информация
* Scroll View за по лесно разглеждане на цялото съдържание на страницата
* Custom Background за Edit text – овете и Vertical Layout
* Радио група, CheckBoxes, Радио бутони и също Rich Text Area
* Добавен е и custom ripple\_effect за бутоните
* Custom направена сянка – тя е направена отново чрез xml файл.
* Custom font – Оригиналният font използван от Андроид е импортиран в проекта като tiff файл.
* Custom изображения – те са обработвани с програмата Photoshop за да служат за дизайна на приложението



Фигура 4. GUI – стара версия само за референци

Text View

Edit Text

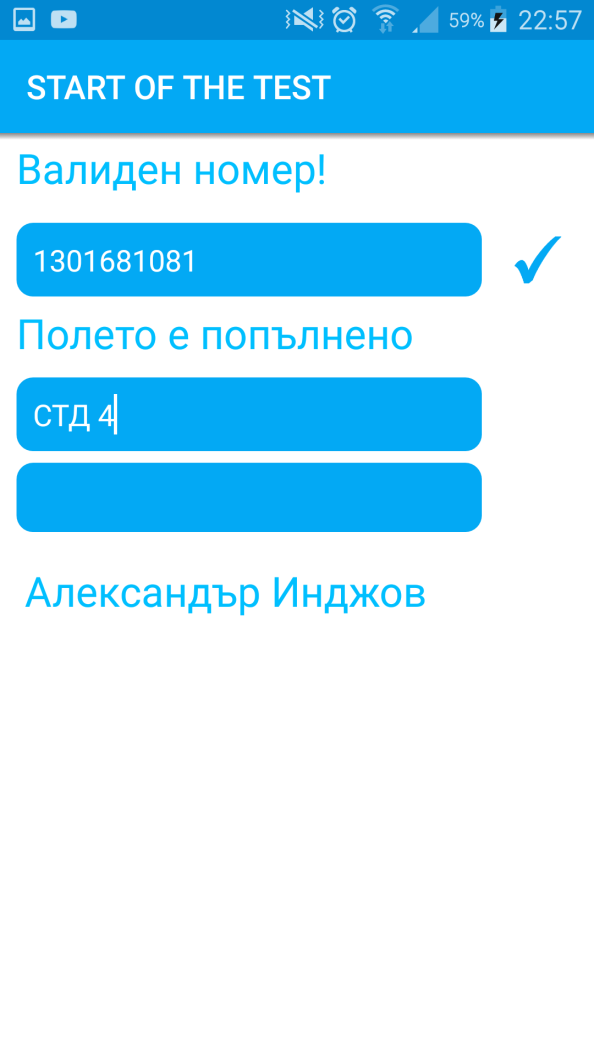
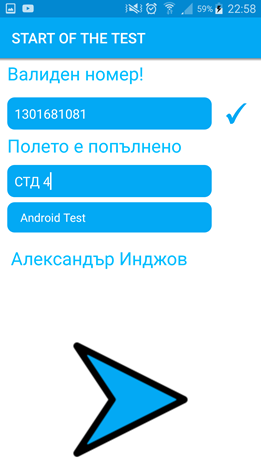


Image Button



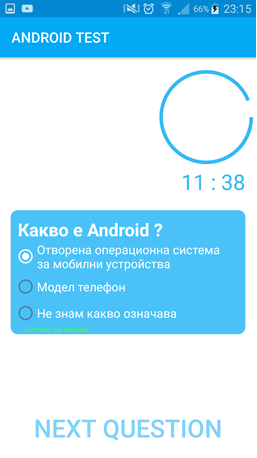
Фигура 5. GUI – настояща версия

За приложението и темата са използвани цветовете записани в colors.xml, а кода за color.xml е както следва :

*<?***xml version=“1.0“ encoding=“utf-8“***?>*<**resources**>  
 <**color name=“colorPrimary“**>#03A9F4</**color**>  
 <**color name=“colorPrimaryDark“**>#0288D1</**color**>  
 <**color name=“colorAccent“**>#FFFFFF</**color**>  
</**resources**>

А специално самата теме е Material.Light.DarkActionBar и е поставена в styles.xml, а описателния код за нея е :

<**style name="Material.Light.DarkActionBar" parent="AppTheme"**>  
 *<!-- Customize your theme here. -->* <**item name="colorPrimary"**>@color/colorPrimary</**item**>  
 <**item name="colorPrimaryDark"**>@color/colorPrimaryDark</**item**>  
 <**item name="colorAccent"**>@color/colorAccent</**item**>  
</**style**>

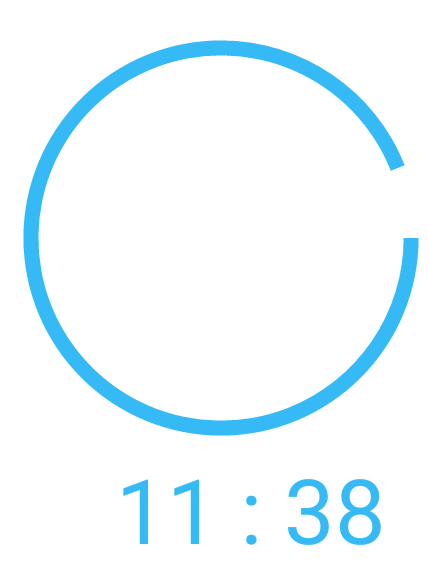
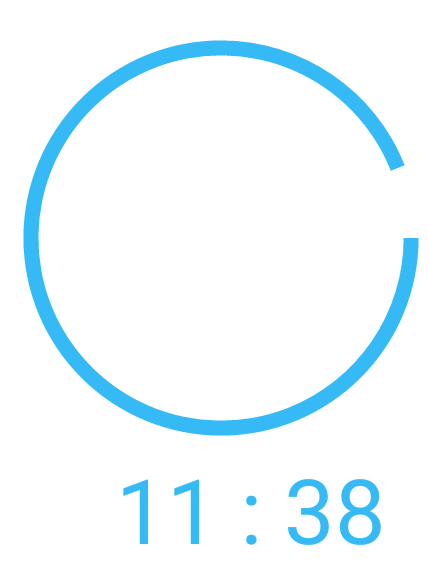
**

drop\_shadow

Custom Background

Circular Progress

Bar



Кода за направта на този ProgressBar се намира в папка drawables под името circular.xml и е следният :

*<?***xml version="1.0" encoding="utf-8"***?>*<**shape  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:innerRadiusRatio="2.3"  
 android:shape="ring"  
 android:thickness="5dp"  
 android:useLevel="true"**>  
 <**solid android:color="#03A9F4"** />  
</**shape**>

Кода за направта на този Custom Background се намира в папка drawables под името shape.xml и е следният :

*<?***xml version="1.0" encoding="utf-8"** *?>*<**shape xmlns:android="http://schemas.android.com/apk/res/android"**>  
 <**solid android:color="#03A9F4"** />  
 <**corners android:radius="10dp"**/>  
</**shape**>

За ripple\_ effect е използван много прост xml код, обаче самия ripple\_effect се потдържа само от по новите версии на андроид от LEVEL 20 нагоре

Кода за ripple\_effect е следният :

*<?***xml version="1.0" encoding="utf-8"***?>*<**ripple xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:color="#03A9F4"  
 tools:targetApi="lollipop"**>  
 <**item android:id="@android:id/mask"**>  
 <**shape android:shape="rectangle"**>  
 <**solid android:color="#03A9F4"** />  
 </**shape**>  
 </**item**>  
</**ripple**>

Относно сянката, Тя е направена с помощта на нов xml файл, както беше споменато по горе в спецификацията на име drop\_shadol.xml. Той се намира в папката drowables и неговият код е съответно :

*<?***xml version="1.0" encoding="utf-8"***?>*<**shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle"**>  
 <**gradient  
 android:startColor="@android:color/transparent"  
 android:endColor="#44000000"  
 android:angle="90"**/>  
</**shape**>

## Хардуерни Интерфейси – неизползваемо

Първоначално за да приложението ще е необходимо самото устройство на което да го инсталирате и използвате. За препоръка не само да имате телефон с андроид операционна система, а телефон с по-голяма версия андроид операционна система.

При възможност, че приложението не може да се набави от студента чрез изтегляна на необходимата версия тогава ще му се наложи чрез USB или Bluethoot да се свърже с преносимия си компютър за да го качи на телефона и съответно да го инсталира и използва

## Софтуерни Интерфейси -neizpolzwaemo

Това приложение, което е представено тук е клиентската част от бъдещо клиент-сървър приложение, като сървърната част ще бъде написана на JAVA и тя ще представлява един SERVLET. Цялостната база ще бъде изнесена за ползване от SERVLET-А

Той от своя страна може да изпрати обратно отговор(response) на заявката изпратена от клиента (request) за да се извърши комуникация.

По този начин се получава финалното клиент – сървър приложение, а за комуникацията помежду комуникацията двете страни, ще се обсъди малко по-надолу в точка 3.4

## Комуникационни Интерфейси

Сървърната част от това приложение, както казахме ще съдържа

в себе си същата база данни, както в приложението, само че тя ще

бъде разширена, за да може приложението да има по-голяма възможност за разнообразие. Под това се има предвид, че ще има възможност приложението да получава за отговор от сървъра различни и разнообразни въпроси на студентите. За да се поддържа разнообразието.

От гледна точка на комуникацията клиента ще я извършва с сървъра чрез интернет достъп, който е разрешен на клиента и за целта е

направен forwarding на използваният модем( TP-Link ). 

Като се предвижда комуникацията да се осъществи с помощта на предефинирания клас HttpClient, който ще извършва комуникацията със SERVLET-A. От гледна точка на SERLET-А той ще се погрижи да връща заявките, които клиента ( в случая приложението ) е направило до него.

# Системни функционалности

* Логин за студенти в програмата
* Избиране на тест
* Решаване на тест
* Записване на резултатите в базата данни

## Логин за студенти в програмата

### Описание и приоритет

За тази цел студентите трябва да въведат факултетният си номер в Edit Text поле, което се валидира и ако действително съществува този студент в базата данни се показва и накрая се пази резултат за неговия факултетен номер.Тази функционалност е с голям приоритет за работата.

### Последователност от запитвания/отговор

Запитване: Студент иска да влезе в програмата

Отговор: В отговор на студента системата, чрез допълнителни проверки разбира, съществува ли студента или не.

**4.1. Функционални изисквания**

REQ-1:Системата проверява в базата за факултетният

номера

REQ-2:Системата ще валидира въведеният факултетни

номер, ако той е правилен.

## Избиране на тест

### Описание и приоритет

### Вече веднъж влезнал в програмата следващата му задача е да изберат от падащ списък своя тест за решаване.Като Приоритета на тази задача е също от много високо ниво за студентите.

### Последователност от запитвания/отговор

Запитване: Студента иска да си избере тест за решаване, като натиска един от многото избори от падащия списък.

Отговор: За избора на студента се показва теста, който е избрал да решава.

### Функционални изисквания

REQ-3:Системата подбира въпроси на свободен принцип, за конкретния тест

REQ-4: Системата подготвя изгледа, който трябва да се

покаже на базата на избрания от студента тест.

## Решаване на тест

### Описание и приоритет

След като вече е избран тест за решаване, на студентите се предоставя възможността да го реши. Тук трябва също да се спомене, че въпросите за теста се генерирани на произволен принцип от системата за съответният тест избран от студентите.Въпросите освен това кореспондират с темата на теста и могат да бъдат въпроси с един правилен отговор, с много правилни отговори или отворен въпрос ( въпрос със текст,който студента трябва да попълни )

### Последователност от запитвания/отговор

Запитване: Студентът трябва да реши избрания тест, като избере отговори на зададените въпроси.

Отговор: Накрая системата му връща изглед в който може да види краен резултат.

### Функционални изисквания

REQ-5: Системата оценява всеки отговор в реално време.

Т.Е. резултатите се виждат от студента на момента на

даване на отговор.

REQ-6: След всеки избран отговор системат раздава точки

за правилните отговори ( те са 5 точки, а грешните 0 )

## Записване на резултатите в базата данни

### 4.4.1 Описание и приоритет

След като студентът е решил съответния тест идва времето той да си получи резултата от изпита и крайните точки от всички въпроси взети заедно. Това става с помощта на изглед в който тази информация може да се види, при натискането на бутон за информация, като съответно тази информация ще се запише в базата данни.

### Последователност от запитвания/отговор

Запитване: След като студента реши теста данните трябва да се запишат в таблица в базата данни

Отговор : Въз основа на желанията на студентът, системата записва данните в базата данни

### Функционални изисквания

REQ-7: Системата записва желаните данни веднага след решаването на теста, като се прилагат алгоритми ( insert query ) за случая.

# Други нефункционални изисквания

## Изисквания за изпълнениe

функционалност за ранкинг система ще бъде добавена в бъдеще, за да може стидентите да са мотивирани и когато използват апликацията да могат да получават изгода и поущрение за положените усилия. Било то под формата на допълнителни точки или освобождение от изпити. Тази ранкинг система ще работи на базата на това, колко добре са се справили студентите на решения тест. Колкото повече точки толкова е по – голям шансът им да попаднат в този списък и да се отличат от останалите студенти.

Логин за реален преподавател ще бъде имплементиран в сървърната част за да може вече преподавателите да контролират, съдържанието и съответно въпросите за конкретните тестове и съответните им отговори. По този начин ще се даде възможност системата да бъде приложима и за преподаватели освен само за студенти.

Освен това ще се даде възможност освен за добавяне на въпроси към тест, за променяне, търкане. Като цяло на преподавателят ще му се даде пълен контрол над сървърната част.

## Изисквания за безопасност

От гледна точка на безопасността, не са допуснати грешки или бъгове в софтуера, но ако съответно някъде се получи бъг, трябва да се уведоми за него и в следваща версия на приложението този бъг евентуално ще бъде отстранен от приложението.

## Изисквания за сигурност

От гледна точка н сигурността на данните те ще могат да се виждат само и единствено от легални студенти на Пловдивския Университет. Това ще рече, че данните ще са достъпни само и единствено за студенти, чиито факултетни номера вече присъстват в официалната база данни на Пловдивския Университет.

Естествено данните няма да са видими изобщо за хора извън Университета.

Това ще рече, че студенти от други университети няма да имат достъп до това приложение.

Ще се направи допълнителен “back–up” файл, ако се случи нещо непредвидено със системата или използваният оригинал.

Логин в Сървлет-а е ще има допълнителна валидация за своите полета, за по-голяма сигурност, срещу некоректни или объркани данни от страна на преподавателя.

## Характеристики на качеството на софтуера

За приложението се предвижда да се направи FAQ ( Frequently Asked Questions), ако потребителите все още изпитват затруднение с неговото използване. Този форум може да се използва както от студентите така и от преподавателите ако някога срещнат трудност в SERVLET-A , защото в него ще има отделна секция само за препдавателя и отделни въпроси, който да задават за SERVLET- A

# Други изисквания

В предишна версия на клиента по-рано беше показан стартовият екран. Сега тук в тази точка ще бъдат показани и други елементи, които са се ползвали за първоначалната клиентска част на приложението.

Първоначално то се състояло от 5 различни екрана, съответно за 5те въпвоса,като в настоящата версия това е променено и базирано на основата на няколко леаута(Linear Layout(Vertical)) и голямо намаляване на android кода.

Има ситуации в които в процеса на работа се е наложило да се обработват снимки с Photoshop, за да се получи търсеният цвят и ефект на първоначалната версия на приложението.

Пример:

* снимка с папка показваща,че това е тест 72874-folder-check
* снимка за таймер. Описание: C:\Users\Sasho Pc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Objects-Timer-icon.png
* снимка за изключване на приложениетоОписание: C:\Users\Sasho Pc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\shutdown.png
* снимка за проверка на факултетен номер Описание: C:\Users\Sasho Pc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\eTMKjnGTn.png
* снимка за проверяване на скалата за оценяване info-512

Самата снимка в предишната версия на приложението е била custom снимка. Тя се поставя в AndroidManifest файла отговарящ за цялото приложение с този код.

android:icon="@mipmap/ic\_penciel"

нова снимка - pencil-xxl

За сегашната версия, естествено тя е променена и напълно отговаряща на дейността на проложението.