

DEVELOPMENT OF THE PLATFORM "ENTERTAINING MATHEMATICS" FOR FIRST GRADE

*Assoc. Prof. PhD Dimitar Minchev, Burgas Free University, mitko@bfu.bg
Milen Donchev, graduate student, mgdonchev@gmail.com*

Abstract: This publication presents the development of a multiplatform mobile application for the platform "Entertaining Mathematics" for the first grade using react native technology.

Keywords: JavaScript, React, React Native, Cross Platform App, Math.

РАЗРАБОТКА НА ПЛАТФОРМА „ЗАНИМАТЕЛНА МАТЕМАТИКА” ЗА ПЪРВИ КЛАС

*доц. д-р Димитър Минчев, Бургаски свободен университет, mitko@bfu.bg
Милен Дончев, дипломант, mgdonchev@gmail.com*

Абстракт: Тази публикация представя разработка на мултиплатформено мобилно приложение за платформа „Занимателна математика” за първи клас, посредством react native технология.

Ключови думи: JavaScript, React, React Native, Cross Platform App, Math.

Увод

Игра в обучението по математика е мощно средство за подтикване на учениците към активна познавателна дейност, повишаване на интереса им към математиката и в резултат на това по-лесно усвояване на учебното съдържание.

Играта в живота на човека има колосално значение и особено голяма е нейната роля в живота на децата. Колкото по-малко е детето, толкова по-голямо възпитателно и развиващо значение има тя в неговия живот. В първи клас децата са все още подвластни на желанието и потребността от игра и затова педагогът в начален етап широко я прилага в своята учебна дейност. Използването на игрите за целите на образованието и за по-плавен преход от преимуществено игрова дейност към учебна може да улесни възприемането на знанията по математика.

Ранната училищна възраст е благоприятен период за умствено развитие при децата, за усвояване от тях на научни знания, развиване на познавателна активност. Математиката е наука, която си служи с много понятия, математическият език е ограничен в точна рамка, всичко е ясно и конкретно. Както детската игра, така и математиката е процес на действие.

Процесът на самоподготовка е съпътстван от трудности и би било педагогическа грешка игровите дейности да се отстранят напълно от училищния живот на ученика. Децата не се боят от трудности и се стремят към изява и затова с увереност може да се каже, че именно времето на самоподготовка на децата е прекрасна възможност за прилагане на математически игри за затвърдяване на умения и навици на малките ученици.

Отдавна са известни предимствата в разработката на приложения за игри и те са изключително подходящи за постигане на целите на обучението, формиране у децата мотивация за учене, самоорганизация и саморазвитие.

Обучаващата игра е дейност, при която, играейки, децата се учат. Тя е утвърдено в педагогическата теория и практика средство за уточняване, разширяване и затвърдяване на знанията. За да може една игра да се трансформира в обучаваща, тя задължително трябва да притежава следните елементи: игрови замисъл, обучаваща цел, игрово действие и правила. Точно тези елементи правят образователната игра привлекателна и желана емоционална дейност. Игрите с математическа насоченост са обучаващи игри, защото е характерно, че игровият процес се съпровожда с усвояване от участниците на съдържанието на обучението.

Играта, ученето и трудът се явяват основните видове човешка дейност. При това играта готви детето както за учене, така и за труд, самата тя се явява едновременно и учене, и труд. Дълбоко заблуждение е да се смята, че играта е само забавление.

При създаване на игри математическата страна от съдържанието на играта винаги трябва да се извежда на преден план. Само тогава играта ще изпълни своята роля за математическото развитие на децата и за възпитаването на интерес към математиката.

Целта на разработване на това приложение е мобилното приложение да се използва за обучение по математика в начален етап на образование, който да развие интелектуалните умения на учениците с помощта на информационните технологии. Идеята е традиционното обучение по математика да се допълва от възможностите на мобилно приложение за затвърждаване на знанията по математика в първи клас, чрез възможността приложението да генерира различни математически задачи, които да осигурят трайност на математическите знания, мотивация за учене и повишен интерес на учениците към предмета математика. Използването на подобен софтуер може да облекчи работата на учителя, като даде възможност и самостоятелно учениците да проверят своите знания.

В приложението се дава възможност да се затвърдят практическите умения на ученика да извършва операции събиране и изваждане на числа до десет. Използван е принципа на решаване на задачи от тип уравнение с едно неизвестно, като ученика трябва да познае неизвестно число в математически операции като събиране и изваждане. Отброяването на верни и грешни отговори дава информация както за нивото на знание на ученика, така и може да го стимулира за подобряване на неговите знания и стремеж да покаже по-добри резултати.

Платформата на която е написан кода на това приложение е React Native. React Native е иновативна технология в мобилната разработка. Това е рамката, позволяваща производството на динамични мобилни приложения за Android и iOS . Приложенията React Native се оказват бързи като производителност, интуитивни като дизайн, удобни за използване и много други. Разгледани са плюсовете и минусите на използваната платформа, сравнена е с други водещи платформи за мобилни приложения, и са разгледани някои примерни приложения които използват React Native като база за тяхното създаване.

Обзор на игрите в обучението по математика

Преподаването на математика играе важна роля на всички съвременни етапи на образованието, още от предучилищното до висшето образование.

„Математиката е кралицата на науките, а аритметиката е кралицата на математиката.“
Карл Фридрих Гаус

Думите на великия учен се потвърждават от самия живот: без овладяване на математически знания, успешното и пълноценно съществуване на съвременния човек е немислимо. Тя ни заобикаля навсякъде: време и пространство, броене и форма - всичко това е математика.

Една от целите на образователни институции е формирането у децата на първоначални математически представи и понятия, способността да се ориентират в абстрактния свят на числа, количества и периоди от време, които са трудни за разбиране от децата. Работата по обучението на децата по математика в първи клас се извършва последователно и целенасочено, като от година на година се усложнява, което се отразява и в образователните програми.

По-голям ефект при изучаването на математика от децата може да се постигне чрез

Приспособяването на образованието към информационния век не се изразява с едно просто действие за осъвременяване на учебното пространство, а налага пълна промяна на съдържанието, методите и целите на образователната система. То се изразява в цялостна образователна реформа, включително и в промяна на човешкото мислене. В отговор на необходимостта от модернизирани и хармонизирани образователната ни система със световните и европейските стандарти са приетите с ПМС № 81 Национална стратегия и План за действие за въвеждане на информационните и комуникационните технологии в българските училища (МТС, МОМН, 2005; Министерски съвет на Република България, 2011), а впоследствие и Националната програма „ИКТ в училище“ (МОМН, 2012).

Целта за интегрирането на ИТ в обучението на учениците от I – IV клас е свързана с разширяване и обогатяване средата на обучение, подобряване на обучението чрез ефективно приложение на ИКТ, прилагане на интерактивни методи и стратегии на преподаване и учене с помощта на ИКТ, подобряване на съществуващите статични форми на учебни материали и включване на динамични мултимедийни разработки, използване на електронни и мултимедийни учебни материали в процеса на преподаване в клас, използване на тестове и материали, които да са алтернатива на конвенционалната форма на обучение.

Използването на ИТ в обучението по учебните предмети от задължителната подготовка в I – IV клас е насочено към подпомагане усвояването на учебното съдържание от учениците. Основните образователни задачи за часа може да бъдат изведени от окачваните резултати от обучението по учебния предмет, определени в учебната програма. Като допълнителна задача може да бъде дефинирана такава, свързана с активното използване на ИКТ в обучението по съответния учебен предмет.

Избор и обосновка на използваните технологии

React Native е гореща тема в света на мобилните разработки от няколко години. Нищо чудно – той завладя света на технологиите, предлагайки начин за разработване на мобилни приложения за iOS и Android едновременно. React Native е единна рамка за управление на всички тях. React Native е успешно възприет от стотици компании по

целия свят, включително Uber, Microsoft и Facebook, и се използва от много други [5] . Именно тази платформа използвам при разработката на мобилното приложение.

Какво е React Native?

React Native (известен също като RN) е популярна рамка за мобилни приложения, базирана на JavaScript, която ви позволява да създавате собствени мобилни приложения за iOS и Android. Рамката позволява да се създават приложения за различни платформи, като се използва една и съща кодова база.

React Native е пуснат за първи път от Facebook като проект с отворен код през 2015 г. Само за няколко години той се превърща в едно от най-добрите решения, използвани за мобилно развитие. Разработката на React Native се използва за запазване на някои от водещите мобилни приложения в света, включително Instagram, Facebook и Skype.

Какво прави React Native уникален?

Разликата между React Native и други крос-платформени решения за разработка (като Cordova и PhoneGap) е, че React Native не изобразява WebView в своя код. Той работи на реални, собствени изгледи и компоненти. Това е една от причините за впечатляващият успех на React Native.

Примерни приложения, създадени с React Native

Facebook

Facebook е едно от най-популярните React Native приложения и не случайно на първо място , тъй като той роди този език за програмиране и е основната движеща сила зад неговото развитие. Facebook искаше да донесе всички предимства на уеб разработката на мобилни устройства, като бърза итерация и обединен екип за разработка на продукти, и така се роди React Native. Компанията го използва, за да разработи свое собствено приложение Ads Manager за iOS и Android, като и двете версии са създадени от един и същ екип за разработка.

Skype

Skype е друг добър пример за мобилно приложение на React Native. През 2017 г. Skype обявява, че създава чисто ново приложение, базирано на React Native. Това предизвика много вълнение сред потребителите, тъй като по-старата версия страдеше от няколко проблема. Новото приложение е изцяло преработено, от иконите до новия интерфейс за съобщения, който вече има три секции за разговори: търсене, чат и писане. Microsoft, която притежава Skype, използва React Native не само в мобилното приложение, но и в настолната версия на платформата.

Instagram

Instagram реши да интегрира React Native в тяхното съществуващо родно приложение, като започна с изгледа Push Notification, който излезе като WebView. За щастие нямаше нужда от създаване на навигационна инфраструктура, тъй като потребителският интерфейс беше достатъчно прост, за да се управлява без него. Използването на React Native е позволило на екипите за разработка да увеличат скоростта на разработка с 85 - 99%.[8]

Walmart

Друг интересен пример за React Native са приложенията на Walmart за iOS и Android. Американското приложение за пазаруване и хранителни стоки е известно със своите смели технологични решения - и едно от тях е , че напълно пренаписа мобилните си приложения в React Native. След преминаване към React Native, производителността на приложенията за iOS и Android се подобри значително - почти до естественото

ниво. Деветдесет и пет процента от кодовата база се използва за Android и iOS; Освен това има единен екип, който управлява и разработва и двете приложения.

Sound Cloud Pulse

SoundCloud реши да се обърне към React Native, за да разработи SoundCloud Pulse, приложение за създатели на музика[9]. Въпреки че първоначално са възнамерявали да разработят две отделни собствени приложения (и да започнат с iOS), е било трудно да наемат екип от iOS инженери. Оказва се, че разработването на две отделни приложения би означавало, че няма да могат да поддържат версиите на iOS и Android в синхрон. Затова решават да тестват React Native и разработват прототип на услуга, използвайки рамка за мобилно приложение за различни платформи. Инженерите на SoundCloud са изумени от темпото на работа - прототипите на екрана били написани до края на първата седмица. За тях също било лесно да свържат съществуващите си собствени библиотеки с React Native. Този положителен опит е убедил SoundCloud да използва рамката за бъдещото си приложение.

Shine

Shine е друг чудесен пример за приложение на React Native. Той помага на своите потребители да се справят с ежедневиия стрес чрез медитация, вдъхновяващи статии и др. Когато създателите на Shine за първи път решават да превърнат идеята си в приложение и да я представят на американския пазар, те залагат на iOS (която по това време представлява около половината от пазара на мобилни устройства в САЩ). Те обаче планират да привлекат вниманието на потребителите на Android, ако приложението им за iOS придобие популярност. Това е мястото, където React Native влиза в играта.

UberEats

UberEats е друго приложение, базирано на разработката на React Native. За разлика от приложението Uber, то включва три страни вместо две – ресторанти, партньори за доставка и посетители. Това е наложило създаването на специално табло за управление, което да отчита и ресторантите. Оригиналното табло за управление, създадено специално за мрежата, ограничавало възможността за предаване на важна информация на ресторантите. Освен това липсвало достъп до вградени функции на устройството, като звукови известия, което се отразило негативно на потребителското изживяване. Екипът е бил не достатъчно запознат с Android и iOS, въпреки многото опит в програмирането на React, и така изборът на React Native остава естествен избор. UberEats използвал голям технологичен стек, а React Native е само малка част от него. Разработчиците били доволни от това, което може да се предложи, и са уверени, че то ще може да отговори на техните нужди с нарастването на пазара.

Pinterest

Инженерите на Pinterest признават, че следят React Native и се интересуват от отключването на неговите възможности, откакто рамката дебютира през 2015 г. По това време те вече предлагали на потребителите уеб приложение, базирано на Gestalt, набор от Pinterest UI компоненти с отворен код, React. Тъй като и Gestalt, и React Native са базирани на React, инженерите на Pinterest очаквали, че разработката няма да бъде сложен и отнемащ време процес. Въпреки че никога не са възнамерявали да заменят напълно своето Gestalt приложение с React Native, те тествали дали то може безпроблемно да се интегрира в текущия им технологичен стек.

Предимства на React Native

Разгледахме някои продукти, създадени с React Native, така че нека да преминем към предимствата на разработката на React Native и избора му като решение за изграждане на мобилни приложения.

Повторна употреба на кода - разработка на различни платформи

Възможността за повторно използване на код е най-голямото предимство на React Native и показва, че приложенията могат да работят ефективно на множество платформи. Може да се интегрират 90% от вградената рамка за повторно използване на код за двете операционни системи. Друга предимство е, че кодът на уеб приложение може да се използва за разработване на мобилни приложения, стига и двете да използват React Native. Той също така това ускорява времето за разработка, защото включва предварително разработени компоненти, които са включени в библиотеката с отворен код.

Голяма общност на разработчици

React Native е JavaScript рамка с отворен код, която позволява на разработчиците да допринесат за разработването на рамка, достъпна за всеки. Ако някой разработчик срещне проблем при разработването на приложението, той може да потърси подкрепа от общността. Винаги ще има някой, който може да помогне за решаване на проблема – това също има положителен ефект върху подобряването на уменията по програмиране.

Икономическа ефективност

Друго предимство на разработката на React Native е по-високата рентабилност. Това е така, защото разработчиците могат да използват един и същ код, за да създават както приложения за iOS, така и за Android. Това означава, че не е нужно да се наема два отделни екипа за разработка на iOS и Android, за да завърши даден проект. За създаването му е достатъчен малък екип. Цената за разработване на приложения в React Native е много по-ниска от приложения, създадени с помощта на езици, които не позволяват разработка на различни платформи.

Бърза актуализация

Бързата актуализация позволява на разработчиците да стартират приложението, като го актуализират до нови версии и променят потребителския интерфейс. Промените се виждат незабавно и на разработчика е спестено необходимостта от повторно изграждане на цялото приложение.

Това води до две значителни ползи: спестяване на време – защото програмистите пестят време за компилиране и повишаване на производителността – тъй като те не губят никакво състояние, когато се правят промени в приложението.

Опростен потребителски интерфейс

Разработката на React Native използва React JavaScript за изграждане на интерфейса на приложението, което го прави по-отзивчив и по-бърз с намалено време за зареждане, което подобрява цялостното потребителско изживяване. С реактивен потребителски интерфейс и компонентен подход, рамката е идеална за създаване на приложения както с прост, така и със сложен дизайн.

Бързи приложения

Има съмнения, че собственият код на React може да повлияе негативно на производителността на приложение. Въпреки че JavaScript няма да работи толкова бързо, колкото собствения код, разликата е невидима за човешкото око .

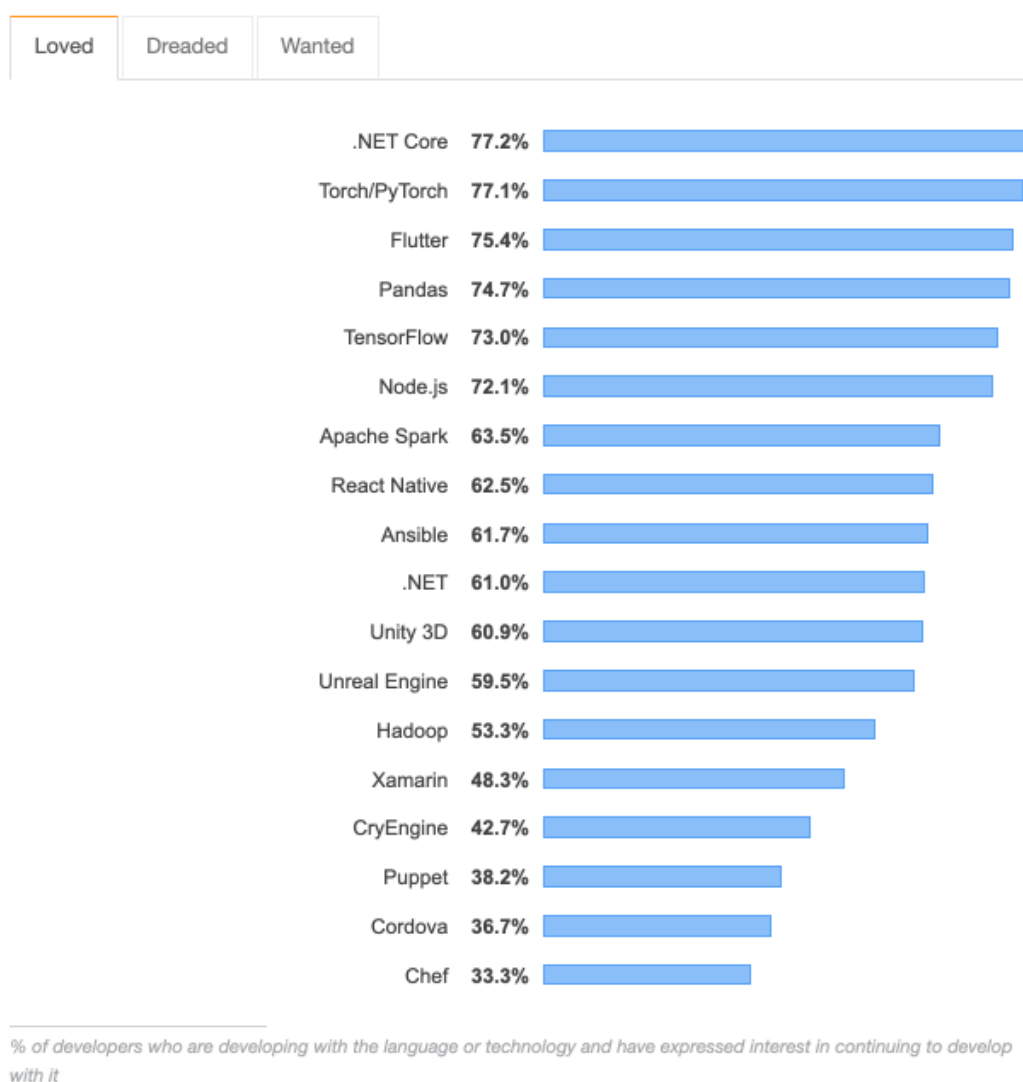
Перспектива

Като се има предвид скоростта, с която платформата завладя пазара, и простият ѝ подход за решаване на проблеми при разработката, бъдещето на React Native за междуплатформени приложения изглежда светло. Въпреки че има няколко недостатъка, скоростта и лекотата на разработка го компенсират повече от това.

Алтернативи на React Native

За да имаме база за сравнение можем да разгледаме някои от алтернативите на React Native. Един от основните му конкуренти е Flutter който ще разгледаме по обстойно. React Native и Flutter са две крос-платформени платформи, разработени от добре известни технологични гиганти: Facebook и Google. И двете са в списъка на най-обичаните рамки, като разработчиците предпочитат Flutter пред React Native.

Most Loved, Dreaded, and Wanted Other Frameworks, Libraries, and Tools



Фиг. 1. React Native Alternatives

Това, че Flutter е предпочитаният вариант за програмистите, той може и да не бъде най-добрият вариант за изграждане на приложения.

Разработка на платформа „Занимателна математика за първи клас“

Инсталиране на необходимите инструменти

Преди да се започне създаването на проекта е необходимо да се инсталират следните инструменти :

- Редактор на кода , като в моя случай използвам Visual Studio Code[18]:
- Изтегляне и инсталиране на Android Studio за Windows [19].
- Инсталиране на Android SDK
- Създаване на пътища за променливи на средата за Java SDK и Android SDK:
- Инсталиране NodeJS за Windows.

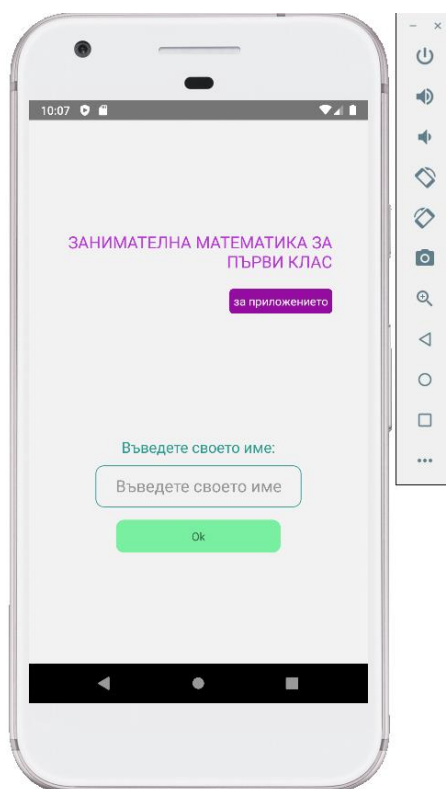
Подготовка на устройството с Android

Необходимо устройство с Android, за да се стартира приложение React Native за Android. Това може да бъде или физическо устройство с Android, или по-често можете да използвате виртуално устройство с Android, което позволява да се емулира устройство с Android на компютъра. Така или иначе ще трябва да подготвите устройството за стартиране на приложения за Android за разработка.

Проектиране на заданието

Заданието ще съдържа пет страници като функционалността им е следната:

Първа страница – тя ще представлява страница за вход в програмата; на която ще има заглавие на приложението „ЗАБАВНА МАТЕМАТИКА ЗА ПЪРВИ КЛАС“; бутон „за приложението“, който отваря последната страница с информация за приложението; текстово поле за въвеждане името на потребителя и бутон „ОК“ за вход в програмата. Която в крайният вариант визуално изглежда така:

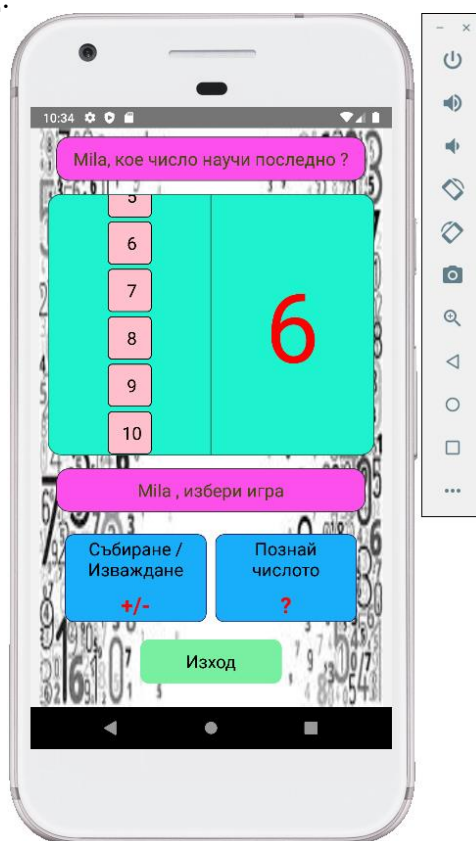


Фиг.2. Welcome Screen

Втора страница – след въвеждане на името на потребителя, приложението преминава към втората страница, на която има следните полета :

- В първото поле е изписано името на детето, и въпрос „кое число научи последно в час“ , като избора на числото в диапазона от 1 до 10 ще определя числата които ще участват в математическите операции включени в задачите от следващите две страници;
- Второто поле е разделено на две части, като в лявата част се избира число от 1 до 10, а в дясната част избраното число се визуализира с по-голям шрифт за да се види избраното число;
- Трето поле подканя детето да избере игра, като му се дават две възможности – единият избор е да избере бутна с надпис „Събиране изваждане +/-“ с който се отваря третата страница на приложението, и другият избор е бутона с надпис „Познай числото ?“ с който се отваря четвъртата страница на приложението;
- Четвъртото поле е бутона „Изход“ чрез натискането на който се връщаме на първата страница;

Страницата има следния вид:



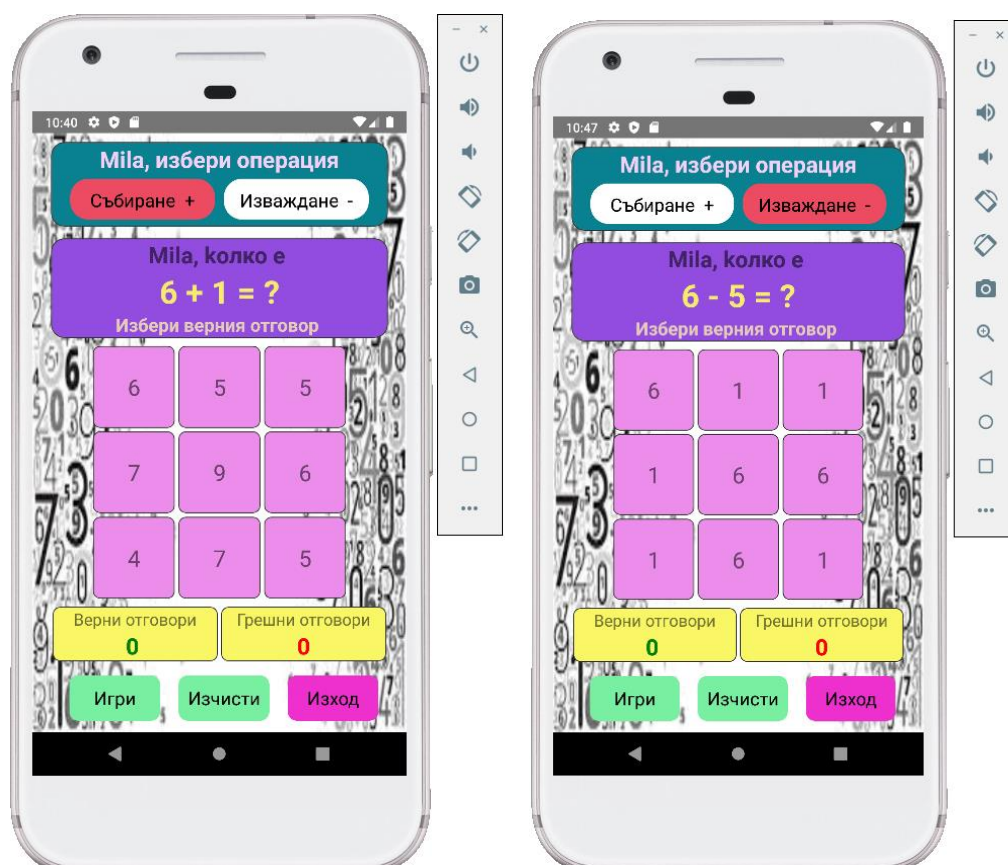
Фиг.3 Choose Number

Трета страница – тази страница се визуализира след избор на бутона „Събиране изваждане +/-“. Страницата има следните полета :

- Първо поле – в него се подканва детето да избере операция между действията събиране и изваждане. Направеният избор оцветява бутона за съответната информация в червено;
- Второ поле – то включва въпрос към детето колко е резултатът от показаната операция, и се подканва детето да избере верния отговор от следващото поле. За всяка операция , независимо дали е събиране или изваждане, числата се генерират случайно;

- Трето поле – изградено е като матрица от клетки 3x3, в които са генерирани девет случайни числа, като поне едно от тях е верният отговор. Детето трябва да избере верният според него отговор като натисне едно от тези девет числа.
- Четвърто поле – то е оформено като две под полета в които се отброяват съответно в ляво верните отговори, и в дясно грешните отговори на избраните отговори на детето.
- Пето поле – в него са бутоните „Игри“ – който втората страница с избор на научно число и вид игра, бутон „Изчисти“ който изчиства резултата за верни и грешни отговори и бутон „Изход“ чрез който се връщаме на първата страница за въвеждане на името на детето.

Полетата на страницата изглеждат така :



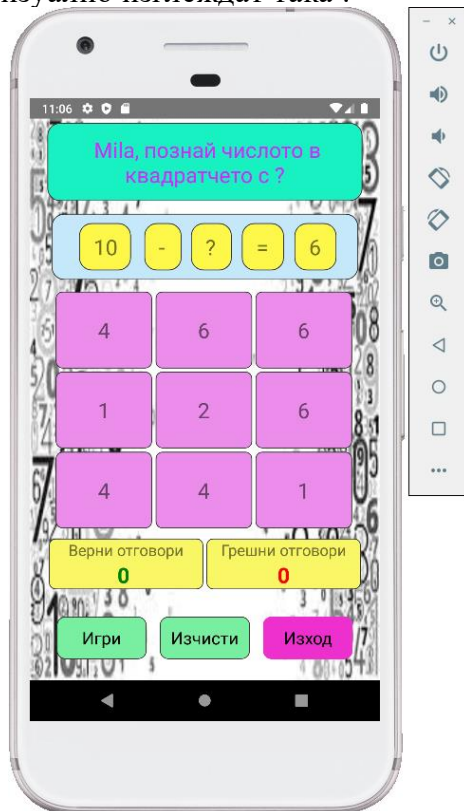
Фиг.4 Answer Math Problem

Четвърта страница – изборът на бутона „Познай числото ?“ от втора страница отваря четвъртата страница на приложението. Идеята тук е да се познае числото, като в същото време по този начин се вдига трудността на математическите операции за детето. Страницата включва следните полета:

- Първо поле – въпрос към детето да познае числото в квадратчето с въпросителен знак „?“;
- Във второто поле на случаен принцип се избира :
- дали операцията да е събиране или изваждане;
- на коя позиция да се визуализира въпросителния знак;
- кои да бъдат числата в съответната математическа операция;
- Третото поле е матрица 3x3 в която са генерирани девет случайни числа, като поне едно от тях е верният отговор на математическата операция

- Четвърто поле – то е оформено като две под полета в като се отброяват съответно в ляво верните отговори , и в дясно грешните отговори на избраните отговори на детето.
- Пето поле – в него са бутоните „Игри“ – който втората страница с избор на научно число и вид игра, бутон „Изчисти“ който изчиства резултата за верни и грешни отговори и бутон „Изход“ чрез който се връщаме на първата страница за въвеждане на името на детето.

Полетата на страницата визуално изглеждат така :

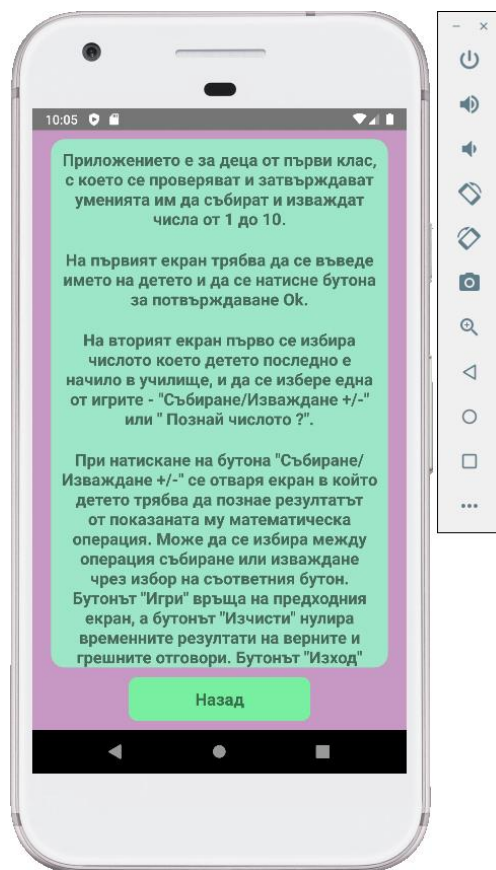


Фиг 5 Guess the Number

Пета страница – в тази страница е записана кратка информация за самото приложение, описани са всички страници и какво трябва да се прави конкретно на всяка от тях. Тя съдържа само две полета , и те са :

- Първо поле – в него е въведена информация за приложението и неговата функционалност. Описани е накратко какво трябва да се прави останалите страници и значението на бутоните ;
- Второ поле – то включва бутон „Назад“ с който се връщаме на първа страница.

Визуално изглежда така :

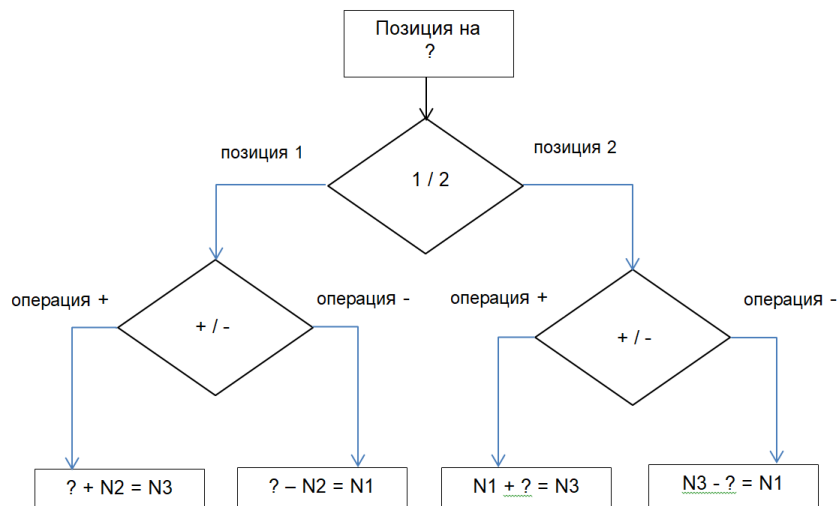


Фиг.6 About Page

Софтуерна разработка на приложението

След проектиране на приложението, се преминава към софтуерната му разработка. Мобилните приложения са съставени от множество екрани. При изграждането на мобилни приложения основната грижа е как да се справим с навигацията на потребителя през приложението. При проектирането на това приложение са предвидени пет екрана с които потребителя ще работи. За целта използвам React Navigation, която е самостоятелна библиотека, и позволява да се внедри функционалност за навигация в приложение на React Native. React Navigation използва това, което се нарича навигатор на стека, за да управлява хронологията на навигацията и представянето на съответния екран въз основа на маршрута, поет от потребител в приложението. Само един екран се показва на потребител в даден момент. За преминава между екрани в приложението ще използвам StackNavigator.

Целта на функцията UravneniePlusMinus е да изпише логически вярно уравнението в зависимост от това каква е операция се е генерирала и каква позиция на въпросителния знак е генериран. Логиката е описана в следващата блок схема :

Фиг. 7. *Uравnenie Plus Minus Page*

За всяка позиция първо се определя какво ще се показва – число или символ, и след това този стринг се изпраща на функцията `Uравnenie()` като параметър, за да се визуализира на определената позиция.

Аналогично и на страницата `PlusMinusPage`, и тук матрицата е от 3×3 клетки с генерираните възможни отговори. Използвана е също компонента `FlatList`, който изобразява елементи от списък в изглед. Данните за клетките се вземат от масива `Cells`. В този случай се използва отново свойствата на компонента – поддръжка на множество колони, в случая три за да се формира матрицата. Функцията `renderItem()` визуализира клетките на матрицата като бутони от тип `TouchableHighlight`. `OnPress()` функцията на тези бутони извиква функцията `getItem()` която беше разгледана по-горе. Числото което се показва във всеки от бутоните се задава. Следващото поле на страницата показва две полета за верни и грешни отговори. Стойността на променливата `answTrue` показва броя на положителните отговори, а стойността на променливата `answFalse` показва броя на отрицателните отговори.

Последната компонента с която завършва файлът `InfoPage.js` е `StyleSheet`. В нея са описани всички стилове използвани за тази страница. Свойството му `hitSlop` определя на какво разстояние извън бутона да се разпознава кликването. `Android_gripple` свойството задава цвят на бутона когато е натиснат. Стилът на бутона `Назад` се определя.

Генериране на приложението

Какво е .apk файл?

Пакетният комплект за Android (APK) е файловият формат на пакета, използван от операционната система Android за разпространение и инсталиране на мобилни приложения. Той е подобен на .exe файла, който имате в Windows OS, .apk файлът е за android.

Генериране на APK

Стъпка 1. Генерирайте хранилище за ключове

Необходимо е генериран от Java ключ за подписване, който е файл за съхранение на ключове, използван за генериране на изпълним двоичен файл на React Native за Android. Създаваме такъв с помощта на `keytool` в терминала със следната команда:

```
keytool -genkey -v -keystore your_key_name.keystore -alias your_key_alias -keyalg RSA -keysize 2048 -validity 10000
```

След като стартираме помощната програма `keytool`, тя ни подканва да въведем парола. Трябва задължително да запомним паролата.

Може да се промени `your_key_name` с всяко име, както и `your_key_alias`. Този ключ използва размер на ключ 2048, вместо 1024 по подразбиране от съображения за сигурност.

Стъпка 2. Добавяне на Keystore към проекта

Първо, се копира файла `your_key_name.keystore` и се поставя в директорията `android/app` в папка на проекта React Native.

Отваряме файла `android/app/build.gradle` и добавяме конфигурацията на хранилището за ключове. Има два начина за конфигуриране на проекта с хранилище за ключове. Първо, общият и незащитен начин:

Това не е добра практика за сигурност, тъй като съхранявате паролата в обикновен текст. Вместо да съхраняваме паролата си за хранилище на ключове в `.gradle` файл, можем да укажем процеса на изграждане да ни подкани за тези пароли, ако изграждаме от командния ред.

Стъпка 3. Освобождение на APK генератора

Избираме директорията в `android`, като изпълняваме: `cd android`

За Windows: `gradlew assembleRelease`

В резултат на това **процесът на създаване на APK е завършен**. Генерирания APK се намира в `android/app/build/outputs/apk/app-release.apk`. След което преименуваме приложението на `first_class.apk`. Това е действителното приложение, което се инсталира на телефона.

Тестване на приложението

Преди да бъде генериран файла `first_class.apk` (Android Package Kit), който е файловият формат на пакета, използван от операционната система Android за разпространение и инсталиране на мобилни приложения, трябва да сме сигурни, че приложението работи без грешки и изпълнява всички необходими функции и алгоритми който са заложиени в него.

Софтуерното тестване е процес на изследване и анализ на софтуера с цел получаване на информация за качеството на продукта и услугата, която се изпитва, и започва още от самото начало на писане на кода. Има разработени много видове тестове за софтуерни приложения [24]. Не винаги методи на тестване на софтуер позволяват еднозначно и напълно да се видят, проявят и установят всички дефекти и правилното функциониране на една програма, така че методите на тестване работят в рамките на формалния процес на проверка върху изследвания или разработвания софтуер.

Тестването на софтуер е изследване, което се прави с цел да се даде информация на всички заинтересовани страни за качеството на продукта или услугата. То трябва да даде обективна и безпристрастна оценка на софтуера, която да позволи на потребителя да прецени и разбере рисковете от употребата му. Техниките за тестване могат да включват изпълнение на програмата с цел откриването на грешки (бъгове). Тестването на софтуер може да се разглежда като процес на валидация и верификация, че дадена програма, приложение или продукт:

- Отговарят на изискванията, залегнали при неговия проект и разработка
- Работи според очакванията
- Може да се въведе в употреба при тези си параметри

- Отговаря на нуждите на всички заинтересувани страни

Тестването на софтуера, в зависимост от избрания метод на тестване, може да се проведе по всяко време в процеса на разработка. В традиционния случай повечето тестове се провеждат след като изискванията са формулирани и етапът на писане на код е приключил. За сравнение, при гъвкавия подход на разработка, тестването е продължителен процес, който обхваща целия период на разработката, а не една отделна фаза накрая. Методологията на тестване е в пряка зависимост от метода на разработка.

Процесът на тестване не може да открие всички грешки в даден софтуер, поради тяхното разнообразие. Основната цел на тестовете е да се открият софтуерни дефекти с цел тяхното отстраняване. Тестването не може да установи дали продуктът ще действа нормално при всички обстоятелства, а по-скоро при какви обстоятелства спира да действа нормално. Обхватът на тестовете може да включва проверка на самия код, неговата работа в различни среди и при различни условия, както и допълнителни аспекти: прави ли това, което се очаква и изисква от него.

Всеки софтуер има целева група клиенти. Например потребителите на игри се различават в изискванията и очакванията си от потребителите на банков софтуер. Следователно всяка организация, която разработва софтуер, се стреми да прецени дали продуктът им ще се приеме от крайните потребители. Тестването на софтуера се стреми да помогне с тази оценка.

Обхват на тестовете

Фундаменталният проблем на тестването на софтуер е невъзможността да се тестват всички възможни входни данни, комбинации от условия и изисквания. Това означава, че разнообразието от дефекти в един софтуер е безкрайно голямо и колкото по-рядко се проявяват те, толкова по-трудно е да се открият чрез тестове. Още повече, че не-операционните аспекти на софтуера (как трябва да бъде, а не какво трябва да прави) са често доста субективни.

Софтуерните разработчици не могат да тестват абсолютно всичко, но те могат да използват комбинирани тестове, за да си осигурят покритието, което им трябва. Комбинираните тестове позволяват да се постигне по-голямо покритие с по-малко тестове. В зависимост от това дали се търси скорост, или дълбочина на тестово проникване може да се използват комбинирани тестове за изграждане на структурирани вариации от тестове.

Принципът на кутията

Методите за тестване на софтуер традиционно се разделят на две групи – метод на отворената кутия (white-box) и метод на затворената кутия (black-box). Основната разлика идва от фокуса на разработчика – дали се интересува повече как работи софтуерът, или дали работи според изискванията.

Отворена кутия

Методът на отворената кутия е метод за тестване на софтуер, който тества вътрешните структури или начина на действие на приложението, но не и неговата функционалност (за разлика от затворената кутия). Той се основава на вътрешния изглед на системата, както и на уменията за програмиране на този, който тества. Специалистът, който тества, избира входни данни, за да проследи обработката им от системата и определя съответните резултати. Това е аналогично на тестването „възми в схема“.

Методът на отворената кутия може да се прилага на ниво модул, на интеграционно и системно ниво, но обикновено се прави на ниво модул. Един от начините е тестване на

пътя. Може да се тества път в рамките на модула, път между отделни модули по време на интеграция, както и между подсистемите по време на тестване на системно ниво. Въпреки че този метод може да разкрие много грешки или проблеми, той не може да открие неизпълнени части на спецификацията или липсващи изисквания.

Методът на отворената кутия работи на ниво изходен код за предотвратяване на всякакви скрити грешки по-късно. Тестовите използват техниките за тестване, както и промяната на обсега условие/решение. Методът използва тези техники като насоки за създаване на среда без грешки чрез проучване на всякакъв чувствителен код. Те изпълняват всеки видим път на изходния код, като целият смисъл на тестването е да се узнае коя линия на кода се изпълнява и да се определи какъв трябва да бъде правилният изход. Това е използвано при самата разработка на приложението за тестване на алгоритмите за разпознаване, дали посоченото число за отговор е вярно.

Тестването тип затворена/черна кутия

При този тест софтуерасе разглежда като „черна кутия“, изучавайки функционалността без каквито и да са познания за вътрешното устройство. Човекът, който извършва теста, е само наясно какво би трябвало дадения софтуер да върши, но не и как го прави. От различните типове тестове от тип черна кутия ще използвам тестване основаващо се на спецификацията .

Изследователско тестване

Изследователско тестване е неформално тестване, проведено от потребител незапознат с програмиране. Целта на това тестване е да изследва приложението и да търси дефекти или грешни резултати, които съществуват в приложението. За целта на потребителя е поставена конкретна задача, която включва попълване на резултатите от използване на двете игри – „Събиране и изваждане +/-“ и „Познай числото ?“. Инструктиран е за всяка игра да направи по 5 верни и 5 грешни отговора, които да запише в таблица и да отбележи вярно ли са отчетени съответните отговори.

За играта „Събиране и изваждане +/-“ резултатите са следните:

Но на теста	Математическа операция	Избран резултат	Отчетен отговор от приложението	Вярно ли е отчитането
1	$5 + 6 = ?$	11	верен	да
2	$7 - 1 = ?$	6	верен	да
3	$5 + 9 = ?$	14	верен	да
4	$10 - 5 = ?$	5	верен	да
5	$6 - 2 = ?$	4	верен	да
6	$4 + 3 = ?$	9	грешен	да
7	$8 + 2 = ?$	5	грешен	да
8	$12 - 7 = ?$	1	грешен	да
9	$2 + 3 = ?$	7	грешен	да
10	$5 - 4 = ?$	6	грешен	да

За играта „Познай числото ?“ резултатите са следните:

Но на теста	Математическа операция	Избран резултат	Отчетен отговор от приложението	Вярно ли е отчитането
1	$7 - ? = 1$	6	верен	да
2	$? + 5 = 9$	4	верен	да
3	$? - 4 = 3$	7	верен	да
4	$8 - ? = 3$	5	верен	да
5	$2 + ? = 12$	10	верен	да

6	$? - 3 = 6$	5	грешен	да
7	$? + 4 = 13$	8	грешен	да
8	$12 - ? = 3$	3	грешен	да
9	$2 + ? = 11$	4	грешен	да
10	$? - 8 = 3$	9	грешен	да

Показаните резултати показват, че теста е минал успешно.

Тестове за използваемост

Този тип тестове са много субективни и зависят от профила на крайния потребител на системата. Няколко съображения трябва да се имат предвид, когато се изпълняват тестовете:

1. Приспособен ли е потребителският интерфейс към профила (образование, социална среда) на крайния потребител?
2. Предоставя ли програмата съдържателни и смислени изходни данни?
3. При възникнали грешки, системата генерира ли ясни и разбираеми съобщения?
4. Системата генерира ли бързо потвърждение на действията на потребителя? Ако някое действие изисква повече време, за да бъде обработено (което е често срещано, когато се достъпва отдалечена система), системата генерира ли съобщение, информиращо потребителя за извършваните операции?
5. Лесно ли се използва системата? Например ако програмата изисква движение през поредица от менюта или опции, връщането в главното меню трябва да става лесно, както и преминаването през различните нива на менюто.

Този тип тестване обикновено се извършва от реални потребители, а разработчика. Разработчика е този, който е създал продукта и следователно не успява да намери по-специфични дефекти, свързани с потребителския опит.

Потокут на приложението се тества, за да се знае дали нов потребител може лесно да разбере приложението или не. По принцип при това тестване се проверява и системната навигация.

За целта готовото приложение е предоставено на няколко ученика от първи клас. Първоначално те са запознати какво трябва да се прави на всеки един екран, като е направена реална демонстрация. Обяснени са функциите на всеки един бутон, и правилата на всяка от игрите. Приложението се оказва интересно за децата, тъй като според тях те не са ползвали подобно до сега. Лесно се ориентират в екраните, и бързо разбраха същността им.

Заклучение

Използването на мобилните приложения като обучаваща игра е още сравнително ново. Това може да бъде в голяма полза на ученика както при прилагане в учебния процес от учителите, така и за самоподготовка в домашни условия под формат на игра.

Във възпитателно-образователния процес, който се реализира в начална училищна възраст, играта намира място и приложение в няколко направления. Като метод на обучение в ситуацията, където подпомага решаването на учебната задача, активизира познавателната дейност, съхранявайки радостта от ученето. Като форма на обучение, която дава възможност да се постигне обучаващ ефект, без да се нарушава игровата ситуация и структурата на играта като специфична дейност. Като самостоятелна

практика на децата тя им дава възможност самостоятелно да избират игрите водени от игрови мотиви.

Приложението лесно може да бъде доразвито например за практическо използване във втори клас за използване на операции като умножение и деление на числа, като разбира се е съобразено с материала който се изучава. Променено в тази посока, то отново може да се използва за натрупване на конкретен опит и рутина при използването на тези математически операции, и да има опция за напреднали.

Използвана литература

1. Новакова, З. Методика на обучението по математика в началните класове, Пд, Хермес, 2004
2. Л. Алексиева, Е. Дамянова, Р. Янчева, Я. Малчева, Специфики и практики при интегриране на икт в българското начално училище
3. Принцип за активност на ученика в обучението - https://theeducationvision.blogspot.com/2014/07/blog-post_7364.html
4. Jumpido – съвременният помощник на началните учители; www.jumpido.com
5. Who is using React Native? <https://reactnative.dev/showcase>
6. Xamarin <https://appfigures.com/resources/insights/microsoft-goes-all-in-on-react-native>
7. Understanding the React Native bridge concept - <https://hackernoon.com/understanding-react-native-bridge-concept-e9526066ddb8>
8. Instagram <https://instagram-engineering.com/react-native-at-instagram-dd828a9a90c7#.3h4wir4zr>
9. Sound Cloud Pulse <https://developers.soundcloud.com/blog/react-native-at-soundcloud>
10. UberEats <https://www.uber.com/en-BG/blog/ubereats-react-native/>
11. Pinterest <https://medium.com/pinterest-engineering/supporting-react-native-at-pinterest-f8c2233f90e6>
12. Stack Overflow <https://stackoverflow.com/tags>
13. GitHub <https://github.com/facebook/react-native>
14. GitHub <https://github.com/facebook/react-native>
15. Reactive programming https://en.wikipedia.org/wiki/Reactive_programming
16. Ionic [https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework))
17. Apache Cordova https://en.wikipedia.org/wiki/Apache_Cordova
18. Visual Studio Code <https://code.visualstudio.com/>
19. Android Studio <https://developer.android.com/studio>
20. NodeJS <https://nodejs.org/en/>
21. Running On Device <https://reactnative.dev/docs/running-on-device>
22. Create and manage virtual devices <https://developer.android.com/studio/run/managing-avds>
23. Metro package for JavaScript <https://facebook.github.io/metro/docs/concepts>
24. Видове тестване на софтуер <https://bg.myservername.com/types-software-testing>