

Изграждане на мобилно приложение към система за управление на дейността по мърчандайзинг – стандартен или хибриден подход

Диляна Димитрова

Mobile Application Development for Management Systems for Merchandising Activities - Native or Hybrid Approach

Dilyana Dimitrova

Abstract

Merchandising organizations are facing the need for a software tool with which they can quickly and easily take into account the data collected on retail store visits by the merchant specialists. To meet this need specialists can use executable on a mobile device software using which to reflect the collected information. It can be categorized according to whether access is through a website or a mobile application. At the second option we consider two types of mobile applications - native and hybrid. The native apps are created for a particular mobile operating system and the hybrid apps combine elements of the other two approaches using standard web technologies. Based on the functionality requirements of the management system for merchandising activities and the mobile device on which it will run we can conclude that the most convenient approach for developing mobile application is the hybrid approach.

Keywords: mobile applications, approaches to mobile development, native mobile application, hybrid mobile application, model view controller, MVC, web site access, accessibility to API, management systems for merchandising activities

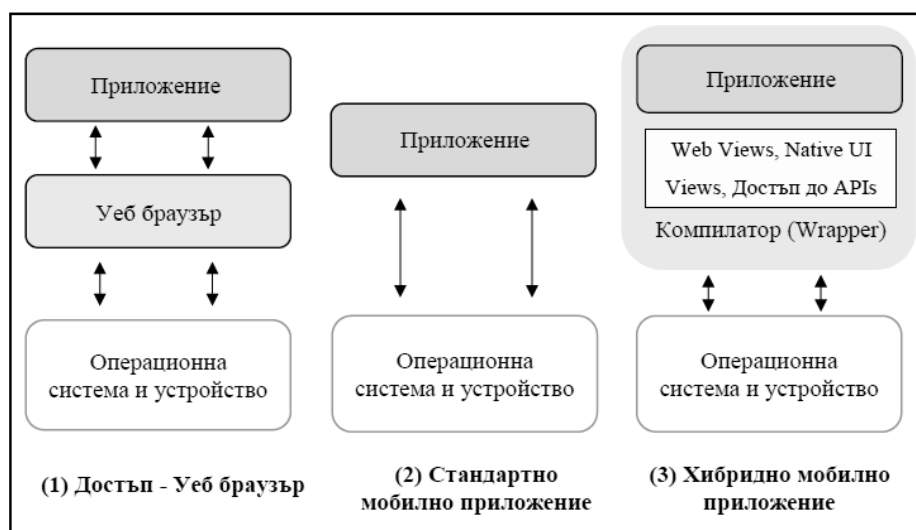
Въведение

Решението дали дадена информационна система да се разработи като стандартен десктоп софтуер или да бъде уеб базирана обикновено се взема в най-ранния етап на проектирането ѝ. Разликата между двата подхода е твърде голяма и изборът на единия обикновено означава окончателен отказ от другия. Проблемът се състои в това, че всеки от двата подхода има свои предимства и недостатъци, поради което този избор не е лесен и еднозначен. Необходимо е да отбележим, че съществуват варианти и за хибридна реализация.

Мобилно софтуерно решение представлява софтуерно приложение, разработено специално за използването му на мобилни устройства като смартфони и планшети. То е проектирано с фокус върху нуждите и ограниченията на подобни устройства, разполага с възможност за директен достъп до всички технологии, налични в приложно-програмния интерфейс от ниско и високо ниво. Мобилните софтуерни решения могат да бъдат категоризирани според това, дали достъпът до тях се осъществява чрез уеб сайт (1) или чрез мобилно приложение. Към втория вариант за достъп могат да бъдат конкретизирани два вида приложения – стандартно мобилно приложение (*native mobile application*) (2) и хибридно мобилно приложение (*hybrid mobile application*) (3), като първите са създадени за конкретна мобилна операционна система (ОС), а вторите – съчетават елементите на двата други подхода.

На фигура 1. разглеждаме трите подхода за достъп. Специфичното при (1) е ролята на уеб браузъра, който работи директно с операционната система на устройството и в чиято среда приложението се стартира. При вариант (2) приложението се изпълнява върху устройството без посредници. В третия случай - вариант (3), е необходим инструмент Web View, който подпомага възпроизвеждането на уеб съдържание в приложението. При него е наличен и достъп до стандартния потребителски интерфейс и достъп до приложно-програмния интерфейс на устройството, който в случая е леко ограничен в сравнение с този, който се предлага при достъп чрез вариант (2). При изпълнението на апликацията и взаимодействието ѝ с ОС, се включва т.нар. Уеб интерпретатор (Wrapper), чрез който

изпълнимият код се „превежда“ към операционната система на мобилното устройство.



Фигура 1. Мобилни софтуерни решения за достъп до приложения

При прилагането на стратегии за внедряване на софтуерни решения за мобилни устройства много организации са изправени пред важно решение, което би повлияло при проектирането на желания продукт. **Необходимо е да отбележим, че процесът по избор на подход за разработване на мобилното софтуерно решение – уеб сайт, стандартно или хибридно приложение, е важен избор, който се влияе от параметри като бюджет, график на проекта, целева аудитория и необходими функционалности на крайния продукт.** Всеки подход носи със себе си предимства и недостатъци, съответно намирането на подходящия, който най-добре адресира нуждите на организацията, може да бъде изключително предизвикателна задача. Основната цел на доклада е да се разгледат предимствата и недостатъците на два от по-подходите – стандартен и хибриден подход за изграждане на мобилно приложение, както и да се опишат различните сценарии за действие в процеса на разработка, внедряване и пускане в експлоатация на крайния продукт, а не толкова да се идентифицира най-добрият подход за развитие, тъй като според нуждите на организацията той може да бъде различен.

Уеб ориентиран достъп чрез мобилен сайт

Първият възможен подход е разработването на мобилна версия на сайт, който би могъл да бъде основният инструмент за отдалечен достъп до системи за управление на дейността по мърчандайзинг през мобилни устройства.

Уеб сайтът предоставя за всички потребители централизиран достъп до единна система. Може да се добави – единна база от данни, което при уеб системите е имплицитно, а при десктоп системите може да бъде направено при необходимост. Уеб сайтовете имат предимството да са достъпни от всеки компютър, което е нова степен на свобода на потребителите. В същото време, те са критично зависими от наличието на връзка с интернет, което рефлектира върху тяхната надеждност – свързана не със самото им функциониране, а с невъзможността за достъп при определени обстоятелства. Тук могат да се добавят и възможните проблеми със сигурността. От една страна, стандартният http протокол не предполага защита на данните по време на преноса им през мрежата. Комуникацията в криптиран вид през https е недостатъчно широко разпространена, понеже изисква разходи за цифров сертификат¹.

¹ Джамбазов В., Уеб базиран потребителски интерфейс, НБУ, 2011

Съвременните мобилни устройства разполагат с мощни браузъри, които поддържат много възможности като HTML5, Cascading Style Sheets 3 (CSS3) и JavaScript. Няколко примера за потенциала на HTML5 включват усъвършенствани компоненти на потребителския интерфейс, достъп до богати типове носители, геолокационни услуги и възможност за работа без постоянна връзка с интернет. С помощта на тези функции и още много други разработчиците са в състояние да създадат по-сложни приложения, без нужда да използват други технологии освен веб технологии. Налична е широка гама от възможности за разработка на такива сайтове. В изследването ще бъде разгледан хибридният подход за разработка на приложения, като ще приложим кратко сравнение между продукт, разработен на негова основа, и мобилен веб сайт. Основната причина за сравнение на тези два подхода е сходността в някои основни характеристики като възможност за поддържане на мултиплатформеност, скорост на пускане в експлоатация, необходими инструменти и познания при програмиране. В таблица 1. са представени характеристики като: инструменти и знание, изпълнение на крайния продукт, възможност за достъп до продукта, както и цялостното представяне.

Таблица 1. Сравнение на четири основни характеристики между мобилно приложение (хибридно) и мобилен веб сайт

Характеристика	Мобилни веб сайтове	Мобилни приложения
Инструменти и знание	Изцяло написани на HTML, CSS и JavaScript	Изцяло написани на HTML, CSS и JavaScript
Изпълнение	Достъп основно чрез конкретен линк (URL)	Възможност за иконка (като при стандартно приложение)
Потребителска ориентация	Навигиране през страниците, визуализация на статични данни	Интерактивен интерфейс, touch-friendly
Цялостно представяне	Целият код се изпълнява на сървъра, а не на локалното устройство, не позволява работа без достъп до интернет	Възможност за работа в offline режим, адаптивно според устройството

Едно от най-съществените предимства при използването на веб базирания подход е неговата мултиплатформена поддръжка и ниска себестойност на развитие и обслужване. Повечето доставчици на мобилни устройства използват един и същ модул за визуализация в своите браузъри - проект WebKit с отворен код, водени главно от производителите Google и Apple, който осигурява всеобхватното прилагане на HTML5. Тъй като кодът на заявление е написан на стандартни веб езици, които са съвместими с WebKit-овете, едно единствено приложение осигурява еднакъв достъп от различни устройства и операционни системи, което го прави мултиплатформен по подразбиране.

За разлика от стандартните мобилни приложенията, които са независимо изпълними, работещи пряко с операционната система, мобилните веб сайтове работят в рамките на браузъра. Той сам по себе си е стандартно приложение, което има директен достъп до приложно-програмните интерфейси на ОС. Но само ограничен брой от тези APIs са налични за използване. Докато native приложенията имат пълен достъп до устройството, за веб приложенията много функции са частично достъпни или не са достъпни въобще. Изхождайки от основната характеристика за възможност за достъп в offline режим и предвид гореизложеното сравнение, можем да заключим, че достъпът до система за управление на дейността по мърчандайзинг през веб сайт не е достатъчно задоволителен, което довежда до ориентация за използване на останалите два предложени подхода – стандартен или хибриден.

Достъп през стандартно мобилно приложение (native mobile application)

Вторият разглеждан подход ще бъде разработването на стандартно мобилно приложение (*native mobile application*), което се инсталира и впоследствие се изпълнява самостоятелно на мобилни устройства с различни операционни системи.

Мобилните приложения имат двоични изпълними файлове, които се инсталират на устройството. Самият процес на инсталиране може да се инициира или от потребителя, или от IT отдела на организацията, в която мобилното софтуерно решение се внедрява и поддържа. Необходимо е да отбележим, че днес най-популярният начин за изтегляне на приложение си остава посещението на виртуален магазин като App Store на Apple, Google Play на Android или Windows Store на Windows Phone, но съществуват и други методи. При инициализация, приложението взаимодейства с мобилната операционна система без посредник. Всички програмни интерфейси, предоставени от доставчика на ОС са достъпни безплатно. В много случаи има уникални характеристики и функции, типични за конкретната мобилна операционна система.

При създаването на приложението, разработчиците трябва да пишат код и да създават допълнителни ресурси като например изображение, аудио сегменти или различни деклариращи файлове към ОС. За да се създаде изпълним програмен код в двоичен вид, който да подлежи на пакетирание с останалата част от ресурсите, той се компилира чрез инструменти, предоставени от производителя на операционната система. Тези инструменти обикновено са в допълнение към други помощни програми и файлове, които накратко се наричат Комплект инструменти за разработка на софтуер (Software development kit /SDK/). Въпреки че процесът на развитие на приложението обикновено е сходен за различните операционни системи, комплектът инструменти е специфичен за всяка една от тях. Тези разлики между платформите водят до един от най-важните недостатъци на текущо разглеждания подход – код, написан за една конкретна мобилна операционна система, не може да се изпълнява на друга, което прави развитието и поддържането за множество платформи продължителен и скъп процес. Необходимо е да се отбележи, че въпреки това този подход е много често използван от множество компании. Тук ключова роля играе приложно-програмният интерфейс (Application Programming Interface). API е интерфейсът на изходния код, който операционната система или нейните библиотеки от ниско ниво предлагат за поддръжката на заявките от приложния софтуер. Приложно-програмният интерфейс предоставя един по-абстрактен и опростен план за разработчика на приложения, който би спестил изучаването на няколко различни слоя от операционната или софтуерната система зад интерфейса. По този начин се постига ефективност и бързина при адаптирането на нови софтуерни технологии. След като инсталираното приложение се стартира от потребителя, то взаимодейства с операционната система с APIs на две нива – ниско и високо².

Именно чрез API от ниско ниво с приложението може да се взаимодейства директно с докосване на екрана или клавиатурата, предоставя се възможност за генериране на графики, връзка с мрежи, получаване на аудио файлове от микрофона на устройството, възпроизвеждане на звук пред високоговорителя или слушалките или работа с камера. Позволява се и достъп до глобалната система за позициониране (GPS) с възможност за получаване на информация за текущи географски координати, както и четене и писане на файлове от твърдия диск или други интерфейси, прикачени към мобилното устройство. В допълнение към предоставяните услуги от ниско ниво, мобилните операционни системи предлагат и услуги от по-високо ниво – процеси като сърфиране в Интернет, управление на календара, контакти, фото и аудио албуми, възможност за провеждане на телефонни разговори или изпращане и получаване на текстови или мултимедийни съобщения. Освен

² Saracut F., Building successful mobile apps, Mobiversal, 2014

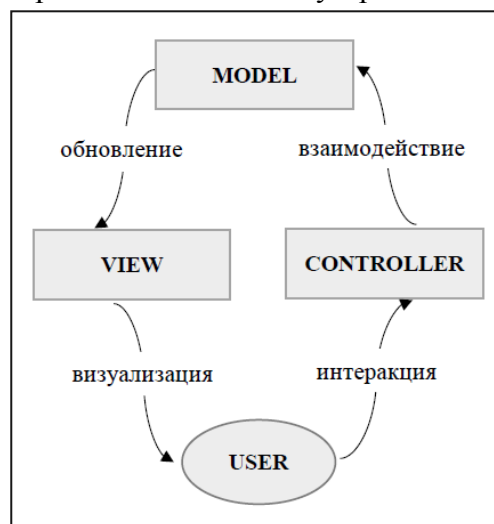
горепосочените APIs от високо ниво, други такива предоставят възможност за сваляне на приложения за достъп до различни базирани в облака услуги, предлагани от доставчика на мобилната платформа.

Всяка мобилна операционна система идва със собствен набор от потребителски компонентни интерфейси, като например бутони, полета за въвеждане, плъзгачи, менюта, разделителни барове, диалогови прозорци и други. Приложения, които се възползват от тези компоненти, наследяват възможностите и функциите на тази конкретна мобилна операционна система, което обикновено води до много лесно и приятно преживяване за потребителя. Важно е да се отбележи, че различните мобилни платформи носят уникални палитри на потребителски интерфейсни компоненти (UI). Тъй като приложението, което е обект на разработка, ще функционира на различни мобилни операционни системи, е необходимо проектантите и разработчиците да бъдат добре запознати с различните потребителски интерфейсни компоненти за всяка мобилна платформа. Въпреки че графичните потребителски интерфейси са специфични според операционната система и добавят сложност и разходи за развитие на множество продукти, тези елементи са единственото средство за създаване на богати мобилни приложения, които да използват пълноценно всички функционалности, предоставяни от съвременните мобилни устройства.

Архитектурата на едно мобилно приложение може да следва шаблона *Model View Controller (MVC)* (фигура 2.). Този модел е основан на разделението на бизнес логиката и правилата от графичния потребителски интерфейс и данните, които ще използва разработваното приложение. Той се състои от три основни елемента - Модел (*Model*), Изглед (*View*) и Контролер (*Controller*).

Контролерът може да изпраща команди към модела, за да актуализира неговото състояние, напр. редактиране или добавяне на съдържание в текущата форма. Може да изпраща също команди към асоциирания изглед за промяна в представянето на изгледа на модела, напр. чрез скролиране на формата.

Моделът уведомява своите асоциирани изгледи и контролери, когато е налице промяна в състоянието му. Това известяване позволява на изгледите да изпратят изходен поток към контролера за продуциране на актуална информация, а контролерите да променят наличния набор от команди. Пасивната имплементация на MVC пропуска тези нотификации, тъй като приложението не се нуждае от тях или софтуерната платформа не ги поддържа. Изгледът от своя страна изисква необходимата информация от модела за генерирането на потребителски интерфейс. Ядрото на приложението се явява моделът. Това е мястото, където се съхраняват обекти. Моделът играе ролята на слой за достъп, извличане и форматиране на данни в цялото приложение. Контролерите са стандартни класове, които получават заявките, инициирани от определено действие на потребителя, обработват програмно данните, подадени от моделите, и зареждат изгледите за изпращане като отговор. Изгледите формират необходимите данни по подходящ начин за визуализация и обикновено биват форматиращи като документни бланки. Елементът Изглед накратко представлява потребителския интерфейс, в който се поставят компоненти като напр. бутони, текстови полета, етикети и т.н. Контролерът и изгледът от своя страна могат да бъдат променяни без да се налагат корекции в модела. Съответно, за един и същ модел могат да бъдат разработени различни интерфейси – например за стандартни мобилни приложения или уеб интерфейс.



Фигура 2. Архитектурен шаблон MVC за програмен дизайн

Достъп през хибридно мобилно приложение (hybrid mobile application)

Хибридният подход съчетава стандартното разработване на мобилни приложения и комбинацията им с актуалните уеб технологии. Използвайки този подход, разработчиците пишат значителни части от кода на приложението, използвайки междуплатформени уеб технологии, чрез които същевременно при необходимост съществува възможност за поддръжка на пряк достъп до приложно-програмния интерфейс на мобилното устройство.

Чрез операционната система на мобилно устройство стандартното приложение използва именно нея, за да създаде вградено HTML ядро, което може да послужи като мост между браузъра и приложно-програмния интерфейс на устройството. Този мост позволява на хибридните приложения да се възползват от всички характеристики, които съвременните устройства предлагат. Разработчиците на приложения могат да изберат да кодират техен собствен мост или да се възползват от готови решения като например PhoneGap библиотека с отворен код, която осигурява JavaScript интерфейс към избрани технологии, работещи на устройствата.

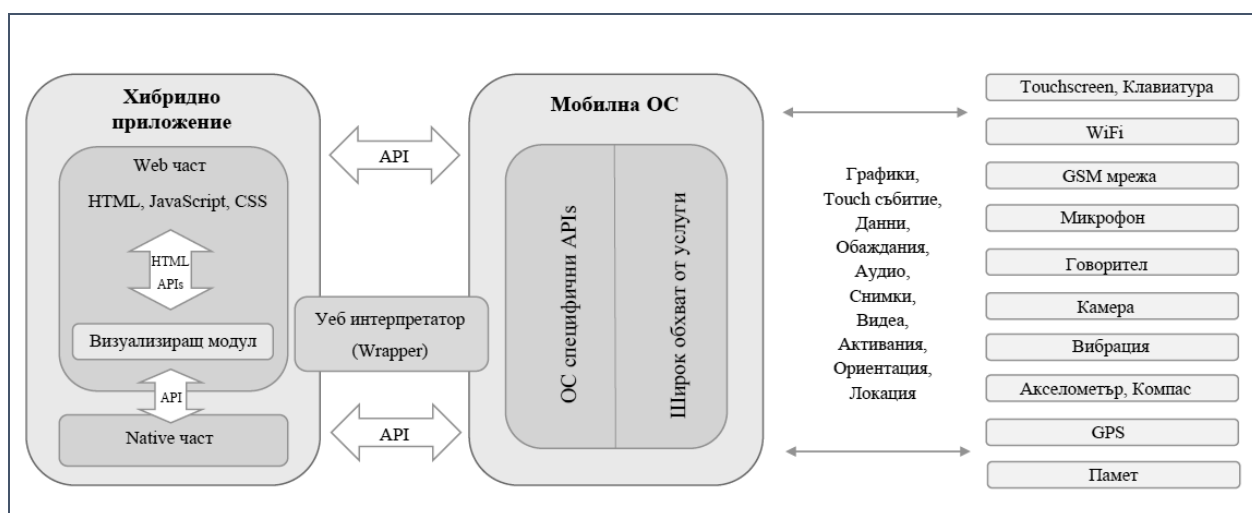
Стандартният код на приложението може да се развива самостоятелно. Някои решения на пазара предоставят този вид контейнер като част от техните продукти. По този начин упълномощават разработчика със средства за усъвършенстване на приложението чрез използване на функциите, налични в устройството, без използването на нищо друго освен уеб езици за програмиране. Тези контейнери могат да се променят в зависимост от уникалните нужди на организацията, за която текущо се разработва хибридно приложение.

Уеб частта на едно хибридно приложение може да бъде или уеб страница, или набор от HTML, JavaScript, CSS и всякакви мултимедийни файлове, окомплектовани в кода на приложението и съхранявани локално на устройството, от което ще се стартира приложението. И двата подхода към хибридно решение притежават своите предимства и недостатъци. HTML код, съхраняван на сървър, дава възможност за въвеждане на минимални промени на приложението без да се преминава през процес на подаването им и последващо одобрение за ъпдейта от мобилните магазини. За съжаление този подход елиминира възможността за използване в т.н. offline режим на устройството и приложението няма да бъде достъпно без наличието връзка с интернет. От друга страна, опаковането на уеб код в самото приложение може да подобри значително ефективността и достъпността, но не предлага възможност за приемане на отдалечени актуализации. Както вече споменахме, необходимо е първо да се премине през процес на одобрение от страна на магазина за дистрибуция на мобилни приложения за конкретната операционна система, след което потребителят да бъде алармиран за наличната актуализация, с което да се инициира ъпдейт на пакета на приложението, инсталирано локално на устройството. Най-доброто от двата подхода може да бъде постигнато чрез комбиниране им – за по-голяма гъвкавост съхранение на HTML ресурсите на уеб сървъра и локално кеширане на данните на мобилното устройство, на което се изпълнява приложението, с което ще се подсигури работата в offline режим.

Хибридните мобилни приложения не се различават от другите приложения, които могат да бъдат намерени в специализираните магазини. Подобно на сайтове в интернет, хибридните мобилни приложения са създадени с използване на уеб технологии като HTML, CSS и JavaScript. Основната разлика се състои в това, че хибридните приложения са инсталирани в стандартно приложение, което използва WebView на мобилната платформа. WebView е библиотека, предоставена от компании като Apple и Google, чрез която е възможно показването на уеб съдържание в стандартни приложения. Чрез показване на уеб съдържание в приложението, WebView помага за подобряване на цялостната работата на потребителя, което означава, че крайните потребители не е необходимо да затварят приложението, за да отворят линк в браузъра. WebView също може да се използва за показване на нещо като страница с правила и условия, което позволява на разработчиците да актуализират страницата без да се налага актуализация на приложението. Това им дава възможност за достъп до технологиите и

функционалностите на устройството, като например акселерометър, камера, контакти и други. Това са инструменти, които често са ограничени до достъп от мобилните браузъри. Освен това, хибридни мобилни приложения могат да включват естествени елементи от потребителския интерфейс в ситуации, при които е необходимо³.

Едно добре разработено хибридно приложение не трябва да изглежда или да се държи по различен начин от неговия стандартен (native) еквивалент. За потребителя няма значение технологията на разработка на приложението. За него е важно то да работи интуитивно, възможно най-бързо и без нужда от превключване на екрани. Хибридните мобилни приложения са изградени по начин, подобен на уеб сайтове. Използва се комбинация от технологии като HTML, CSS и JavaScript. Въпреки това, вместо насочване към браузър, хибридните приложения биват насочени към WebView домакин вътре в родния контейнер. Това дава възможност за достъп до хардуерните възможности на мобилното устройство⁴.



Фигура 3. Примерна архитектура на хибридно приложение⁵

На фигура 3. са представени някои от функционалностите, с които разполага всяко едно мобилно устройство или приложно-програмният интерфейс от ниско и високо ниво. Те биват управлявани от операционната система без значение от нейния основен модел. Хибридните приложения могат да бъдат разделени на две части – web и native. В web частта се включва програмният код, чрез който могат да бъдат инициирани задачи към приложно-програмния интерфейс. Другата част на приложението е отделена за неговата native порция. Чрез Уеб интерпретатора (Wrapper) се осъществява „преводът“ на изпълнимия код към операционната система на използваното устройство.

Сравнителен анализ между мобилните софтуерни решения

Преди разработването на стратегия за развитието на желан софтуерен продукт, е важно да се направи оценка на техническите и нетехнически предимства на хибридните и стандартните приложения. За да се вземе решение, е нужно да се отговори на няколко основни въпроса:

- Към кои мобилни платформи ще бъде насочено приложението?
- Къде и по какъв начин ще се реализира дистрибуцията на приложението?
- Ще бъде ли налице необходимост от използването на функционалностите на

³ Jacobs M., Living on the Edge of Mobile Development, 2015, <<http://cloudcomputing.sys-con.com/node/1719019>>

⁴ Panhale M., Beginning Hybrid Mobile Application Development, 2016

⁵ Gauthier S., Iconum mobile applications development, 2014, <<http://www.slideshare.net/StphaneGauthier1/iconum-mobile-applications-development>>

мобилното устройство?

- Какви са техническите възможности на екипа за проектиране и разработване?

Ако стратегията ще бъде ориентирана към предоставянето на приложение за повече от една платформи, възможностите за избор са няколко. Интернет предлага изключително атрактивно решение за това изискване. Целта е мобилният браузър. Хибридните приложения се поддават добре на този подход, включвайки и приноса на зависимостта си с WebView. От друга страна, стандартните приложения са тясно зависими от SDK на конкретната платформа. Съответно и езиците за програмиране са от значение. В случая, за iOS са Objective-C или Swift, за Android е Java, а за Windows Phone - C #.

Дистрибуцията на крайния продукт може да се изпълни основно в магазините за приложения. В тях могат да се предложат както стандартни, така и хибридни приложения. Изхождайки от факта, че сайтовете са достъпно през уеб браузър, те не могат да бъдат дистрибутирани в подобни магазини. Освен стандартния канал за предлагане на приложения, те могат да бъдат анонсирани и в уеб сайтове с директни линкове, водещи директно до свалянето на мобилното решение.

Стандартно, уеб сайтовете имат ограничени способности за достъп до функционалностите на устройството, достъпът до желания приложно-програмен интерфейс е ограничен. Поради това, много компании се насочват към другите два варианта – разработването на стандартни или хибридни приложения. При последните, достъпът до функционалностите на устройството се случва чрез притурки или т.нар. плъгини (*plugins*). Стандартните приложения са по-подходящи за продукти, които изискват висока графична производителност, напр. игри. За бизнес приложенията полезни са хибридните приложения, напр. някои от тях са Famo.us и Kendo UI. Друг фактор при разработването на стратегия за изграждането на краен продукт е свързана с техническите познания и възможности на екипа. При разработването на стандартни приложения е необходимо програмистите да са запознати с SDK на всяка една от платформите, към които е насочена последващата дистрибуция на крайния продукт. От друга страна, уеб и хибридните приложения са изградени с помощта на уеб технологии като HTML, CSS и JavaScript. Което ограничава значително необходимостта от екип, запознат с множеството платформи на пазара.

На основата на осъществено проучване на трите основни подхода за проектиране и разработване на приложение, което работи на мобилно устройство и нуждите на мърчандайзинг специалистите, може да се обобщи следното: поради възможността мобилното устройство не винаги да разполага с директна връзка с интернет може да се приеме, че уеб ориентиранят подход (или още сайт), няма да бъде подходящ за изпълнение на функциите на мърчандайзерите. За неприложимостта на този подход допринасят и ограничените възможности за достъп до APIs, които в подобни системи биха имали силно присъствие. Макар и тривиална, тази причина натежава при отхвърлянето на този подход. С цел сравнение на ефективността при проектиране, разработка и поддържане на краен софтуерен продукт, разработен с другите два разгледани подхода, ще включим две групи критерии – *основни характеристики (показатели)* и *достъпност до елементите на потребителския интерфейс на устройството*, под което ще функционира приложението.

Анализ по основни показатели

В първата група показатели участват няколко нива (измежду три възможни – ниско, средно и високо) на цялостното представяне на продукта, негова лесна откриваемост, обща достъпност до приложно-програмния интерфейс на устройството, време и цена за проектиране, разработка и поддръжка, възможност за мултиплициране на програмния код и ниво на сигурност. Други разглеждани показатели са свързани с режима на достъп до продукта, негова дистрибуция и използвани програмни езици при разработването. В таблица 2. са представени показателите на двата подхода за изграждане на софтуерен продукт –

стандартно и хибридно приложение.

Таблица 2. Сравнителен анализ на разглежданите подходи за достъп - стандартно и хибридно мобилно приложение

Показател	Стандартно приложение	Хибридно приложение
Цялостно представяне (ниво)	високо	средно
Режим без достъп до интернет	поддържан	поддържан
Дистрибуция	Магазин за приложения	Магазин за приложения
Откриваемост на продукта (ниво)	високо	високо
Мултиплатформена поддръжка	не	да
Достъп до функционалности на устройството (ниво)	високо	средно
Потребителски интерфейс (ниво)	високо	високо
Езици за разработка	стандартни езици за програмиране	уеб/стандартни езици за програмиране
Време за проектиране и разработка (ниво)	високо	средно
Цена за проектиране и разработка (ниво)	високо	средно
Възможност за пренасяне на код (ниво)	ниско	високо
Актуализация на съдържанието (ниво)	високо	средно
Сигурност	високо	високо/средно

На база приложения сравнителен анализ в таблица 2., накратко можем да заключим, че приложение, разработено чрез хибридният подход, предоставя високо ниво на мултиплатформена поддръжка, средно ниво на време за проектиране и разработка и гъвкавост при избор на език за програмиране. От гледна точка на потребителя, предлага достъп независимо от наличието на интернет, достъп до функционалностите на устройството, на което той използва продукта. Считаме че, най-добри значения на показателите по отношение на изискванията на системата за управление на дейността по мърчандайзинг се отчитат при хибридният подход, който предлага лесна мултиплатформена поддръжка на етапи разработване и тестване и възможност за кодиране чрез уеб и стандартни езици за програмиране. За критерии време и цена за изпълнение на продукта е поставена оценка „средна“, но можем да заключим, че те биха били за сметка именно на възможността за изпълнение на един и същ код на няколко мобилни операционни платформи, както и на високо оценената възможност за актуализация на продукта при всички таргетиращи ОС.

Анализ по показатели за достъпност до APIs

Следващата разгледана група показатели към сравнителния анализ е във връзка с възможния достъп до елементи от приложно-програмния интерфейс на мобилното устройство като камера, говорители и микрофон, контакти и календар, достъп до файлове на устройството, GPS, акселометър и жирокоп. В таблица 3. са избрани девет приложно-програмни интерфейси, подбрани според нуждите на система за управление на дейността по мърчандайзинг. Както се вижда и на таблица 3., хибридният подход за разработка не може да предложи достъп до функционалности като микрофон и говорител, докато при стандартното достъпът до разглежданите APIs е пълен⁶. Можем да отбележим, че при изпълнението на задълженията си при посещения по търговските обекти мърчандайзерите не биха имали

⁶ Building Modern Mobile Web Apps, Microsoft – patterns & practices, 2012 <<https://msdn.microsoft.com/en-us/library/hh994907.aspx>>

необходимост от тези два APIs. Отчитайки факта, че биха имали нужда от достъп до контактите, календара, GPS и камерата, вземайки предвид и гореизложените предимства, хибридният подход е най-подходящ при разработване на приложение за достъп до системи за управление на дейността по мърчандайзинг.

Таблица 3. Достъпност до API на стандартно и хибридно мобилно приложение

Достъпност до приложно-програмен интерфейс	Стандартно приложение	Хибридно приложение
Камера	да	да
Говорители	да	не
GPS	да	да
Микрофон	да	не
Контакти/Календар	да	да
Достъп до файлове	да	да
Акселометър	да	да
Жироскоп	да	да
Прикачване на файл	да	да

Заклучение

Въз основа на гореизложеното сравнение на двата основни подхода за достъп през мобилно устройство до системи за управление на дейността по мърчандайзинг, бихме могли да заключим, че инструментът, който най-добре ще спомогне дейността на специалистите, би бил приложение. В случая, разглежданите варианти са два – стандартно мобилно или хибридно приложение. Според изложените предимства във връзка с направения сравнителен анализ на база основни показатели и показатели за достъпност до API, бихме могли да заключим, че от гледна точка на предлагани функционалности и последваща възможност за развитие, насоката за разработване ще бъде ориентирана към хибридните приложения. При тях се отчитат най-добри показатели при мултиплатформено разработване и поддръжка, както и времето и цената за проектиране и привеждане на крайния продукт в експлоатация. Възможностите на хибридните приложения и достъпността им до приложно-програмния интерфейс на мобилните устройства, биха задоволили до голяма степен нуждите на мърчандайзерите при изпълнението на ежедневните им задачи.

Използвана литература

1. Джамбазов В., Уеб базиран потребителски интерфейс, НБУ, 2011
2. Building Modern Mobile Web Apps, Microsoft – patterns & practices, 2012
<<https://msdn.microsoft.com/en-us/library/hh994907.aspx>>
3. Gauthier S., Iconum mobile applications development, 2014,
<<http://www.slideshare.net/StphaneGauthier1/iconum-mobile-applications-development>>
4. Jacobs M., Living on the Edge of Mobile Development, 2015, <<http://cloudcomputing.sys-con.com/node/1719019>>
5. Panhale M., Beginning Hybrid Mobile Application Development, 2016
6. Saracut F., Building successful mobile apps, Mobiversal, 2014

За контакти

докт. Диляна Панчева Димитрова
Икономически университет, гр. Варна
dimitrova.dilqna@gmail.com