

学科紹介スマホアプリの開発

西 村 則 久

The Development of Smartphone Applications Introducing the Department

Norihisa NISHIMURA

要 旨

安田女子大学に設置計画中の新学科を紹介するためのスマホアプリを開発して配布を開始したので報告する。

アプリの画面には学科の関連Webサイトへとジャンプするボタンが複数個あり、ランチャーのような機能を持つと同時に、Webサイトが更新されるとそのことを通知する機能を持つ。

キーワード：スマートフォン、スマホ、Android、スマホアプリ、アプリ開発

I. 開発の経緯

安田女子大学に設置計画中の新学科（以下「新学科」という。）は、コンピュータについてはリテラシーに留まらずある程度広く学ぶ学科であるため、オリジナルのスマホアプリを開発・配布することによって、新学科ではそのようなアプリを作る技術も学べるということをアピールする計画が浮上し、筆者がそれを担当することになった。アプリが持つべき具体的な機能や仕様は示されず、その策定も筆者に委ねられた。

II. 設 計

アプリの開発・配布の目的が宣伝的なものであるため、世間から注目されるべきであることを考えると、高度な技術に基づいた高度な機能を持つアプリや、多くの人に使ってもらえる非常に便利なアプリで、且つ、これまでに無いものを開発するのが望ましいだろうと当初は考えた。しかしそれは、これまでに世界中の開発者がありとあらゆるアプリを開発済みであるから、容易ではない。そのようなアプリが作られるのは、高度な技術や便利なアイデアが先に存在する場合であろう。

そこで別のアプローチとして、既存の新学科関連Webサイトへのアクセスを容易にするという役割を果たすアプリを作ることにした。宣伝のために、訪問者を増やしたいWebサイトとして、安田女子大学・安田女子短期大学の公式サイトと新学科の特設サイトと新学科のブログがあるので、それらの各サイトへワンクリックでジャンプするボタンをアプリの画面上に表示させ、中でも新学科のブログについては更新頻度が高いので、更新があったときには（アプリを起動していないときでも）OS上で通知を行う機能を持たせることにした。

ほかに、アプリの新しいバージョンが公開された後にはバージョンアップを促す機能を持たせた。

Ⅲ. プラットフォーム

スマートフォンの二大プラットフォームに、AndroidとiPhoneがある。両方のプラットフォームで同一内容のアプリを開発し、それぞれを配布するのが望ましいが、今回はAndroid版のみとした。理由は、Android版は開発・配布のために必要な手続きがほとんど無く、開発環境も無料でダウンロードできるのに対し、iPhoneアプリを作るためにはiPhoneのメーカーであるApple社にデベロッパー登録（有償）が必要で、配布の際にも審査（アプリにコンピュータウィルスが混入していないか、内容が公序良俗に反するものでないか、等と思われる）を受けなければならない、費用および時間がかかる。マイナーなバージョンアップを行うにも審査が必要となることなども考慮し、今回はiPhone版については仮に開発するとしても後回しとすることにした。

Ⅳ. 開発環境

本開発には、Androidアプリの公式な開発環境である、Google社のAndroid Studioを用いた。開発言語はJavaである。

AndroidアプリはAndroid Studioのほかに、アシアル社のMonacaやユニティ・テクノロジーズ社のUnityでも開発できると謳われている。

Monacaで開発する場合、HTML5とJavaScriptを用い、比較的簡単に開発できるよう工夫されているうえ、AndroidアプリとiPhoneアプリを共通のソースコードから構築できるようになっている。大抵のことは無料でできるが、実機で実行するネイティブアプリの構築は、無料で利用する場合、一日に行える回数が限られている（エミュレータ等での実行は何回でも行える）。iPhoneアプリは開発用の実機で実行するためにも、また配布するためにも、Apple社への申請ないし審査が必要となる。

Unityは三次元グラフィックスの表示が強力にサポートされており、Monacaと同様にAndroidアプリとiPhoneアプリ、更にはWindowsやその他のアプリまで構築可能である。やはりほとんどのことが無料で行える。ただしiPhoneアプリにおいてはApple社への申請や審査は必要である。

一般的には、同一内容のAndroidアプリとiPhoneアプリを開発する場合、MonacaないしUnityで開発することが有利となるケースは多いであろう。しかし、両開発環境は比較的簡単に開発を行えるための工夫やマルチプラットフォームへの対応がある反面、デバイスに近いOS深部に関わるような処理を組み込むのは概ね難しい。本開発においては、プログラムを常駐させて一定時間ごとに処理を行うことが両環境では困難あるいは不可能であると判断し、Android Studioを選択することとなった。

またUnityに関しては、Unity上で作られたスマホアプリをAndroid Studioで直接開けるプロジェクトへとコンバートしてエクスポートする機能があるので、常駐プログラムに必要な部分のみをAndroid Studioで追加するという方法も可能ではあるが、Unityは三次元グラフィックスのプログラミングに特化された開発環境であり、三次元グラフィックスを必要としないアプリケーション

ョンを開発する場合には利用するメリットはほとんどない。

V. 仕 様

アプリを起動すると初期画面（図1）が表示される。通常の操作は全てこの画面で行えるので、初期画面が本アプリの唯一の画面となっている。



図1 初期画面

画面の左上部には安田女子大学・安田女子短期大学の公式サイト（<http://www.yasuda-u.ac.jp/>）にジャンプするボタン、画面の右上部には新学科の特設サイト（<http://www.yasuda-u.com/zoukei/>）にジャンプするボタン、画面の左下部には新学科のブログ（<http://zoukeidesign.blogspot.jp/>）にジャンプするボタンが表示される。これらのボタンをタップすると、スマートフォン内に標準でインストールされているブラウザが起動し、それぞれのサイトが表示される。画面の右下部にはブログの更新時に通知を行うか否かを設定するスイッチと、通知時に音を鳴らすか否かを設定するスイッチがある。通知をオンにすると、約2時間ごとにバックグラウンド処理でブログの更新確認が行われ、更新されていたときには以下に述べるようなOSに標準の通知方法で通知される。

通知は、通知音とともに、OS画面の最上段にアイコンと文字が表示されることによって行われる（図2）。アイコンと文字は数秒後にはアイコンのみに切り替わる。



図2 通知時のOS画面

通知を消去するまでの間、LEDがオレンジ色に点灯し、通知画面にもアイコンと文字で情報が表示される（図3）。通知画面の文字をタップするとブラウザが起動してブログが表示されるとともに、通知が消去される。



図3 通知画面

なお、この通知と、通知のためにバックグラウンドで行われる更新確認は、本アプリが起動されていないときや他アプリが起動されているときでも通知のスイッチがオンに設定されていれば行われる。ただし深夜から早朝までの間（21時から7時まで）は行われない。

VI. 技術的なポイント

1. プログラムの常駐

本アプリは、アプリを起動していなくても一定時間ごとにブログの更新確認を行えるように、プログラムの一部を常駐させている。

常駐プログラムは、Androidアプリの開発においてニーズが高いと思われるので、単独のクラスを利用して比較的簡単に実現できるのではないかと予想していたが、実際には複数のクラスを組み合わせなければ実現できないようである。

アプリを終了させた後もプログラムの一部を必要に応じて動作させるためには、Serviceクラスを利用する。そして本アプリの場合、常駐プログラムは2時間ごとに更新確認をすれば良く、他のほとんどの時間は何もなくて良いので、CPU資源や電力の浪費防止の観点から、処理スレッドは常に生きているのではなく、更新確認後に処理を終了させ、2時間後にタイマーで起き上がるのがセオリーである。これにはAlarmManagerクラスを利用した。

以上のように、ServiceクラスとAlarmManagerクラスの組み合わせでアプリ終了後も更新確認をバックグラウンドで無駄なく行えるようになったが、これだけだとスマートフォンの電源を切って再起動させた後にはアプリを起動させるまでプログラムが常駐しない。よって、BroadcastReceiverクラスを用いてOSの再起動時にプログラムが自動的に常駐するようにした。

2. ブログ更新の確認方法

当初、ブログの記事作成担当者と話し合っ、て、ブログの更新確認方法として、ブログのWebページファイルのタイムスタンプを記憶し、それが新しくなっていたら記事が更新されたと判断することにしていた。

しかしそれだと、既出の記事を少しだけ修正したときでも通知されてしまうことになる。また、更新でも修正でもないときに、情報システム部門のブログ更新担当者（記事担当者とは別人）が、一見影響無さそうな操作（例えば、ファイルの再アップロード）をしてタイムスタンプが更新されてしまうようなことも稀に起こり得る。ブログ更新担当者に事情を説明しておけば済むことではあるが、事務方であるから異動などの影響で比較的短い間に交替するのが常である。

そもそもファイルのタイムスタンプで更新の確認を行うこと自体があまり信頼性の高くない方法なので、次のような方法で更新確認をするように変更した。

Webページファイルの中味を読み込み、最初の記事のタイトルとそれに続く数件分の記事のタイトルを抽出し、最初の記事のタイトルが前回確認時に無かったものであればブログは更新されていて、前回確認時にも存在していたらブログは更新されていないと判断する。

この方法であれば、記事の本文が少し修正されたような場合には更新されていないと判断されるので、通知されず、都合がよい。また、前回確認時に最初だった記事が何らかの事情で削除されたようなケースでも、通知されないのが都合がよい。ただし、最初の記事のタイトルが修正されたような場合には通知されてしまい都合が悪いが、そのようなことはあまり起こらないと思

れ、タイムスタンプのみによる判断よりも格段に優れていると思われる。

Ⅶ. 今後の課題

1. iPhone版の開発・配布

上述したように本アプリは現時点ではAndroid版のみが開発済みであるので、iPhone版の開発・配布が課題である。

2. 公式マーケットへの登録

本アプリは、安田女子大学内のサーバーからダウンロード・インストールできる¹。ただし、各Android端末の設定により提供元不明のアプリのインストールが許可されていない状態のときは、インストールはブロックされる。よって、インストールするためには一時的にでも提供元不明のアプリのインストールを許可しなければならない。

これを解消するために、アプリを公式マーケットに登録することが今後の課題の一つである。

[2015. 6. 25 受理]

¹ URLはhttp://chocobo.yasuda-u.ac.jp/~nisimura/zoukeidesign_app_install/zoukeidesign.apkである。