

【宣言的 UI を利用したアプリケーションの開発】

Application development using declarative UI

電気系コース 長田 陽太 (OSADA Yota)

1 はじめに

私が通っている学校は、創立から数年しか立っていないため、鍵の貸し借りや時間割の確認のために 1 階に降りなければいけない点など、不便な点が多い。そこで、それらの問題を解決するようなアプリケーションを製作しようと考えた。また、制作を通して、必要になった知識などについて本資料でまとめていく。

2 仕様・要件

- 学校に関する情報をすぐにアクセスできるようなアプリケーション
- 学籍番号を入力して、本当に学生か確認する。
- 確認が取れたら学生情報を読み込む。
- 学生情報など、後から変更される可能性のあるデータはプログラムとは別の場所に保存しておいて、あとから変更できるようにする。
- 学年、コース、進路などに応じて、表示内容を変更する。
- UI デザインについては各種ガイドラインに則る。
- なるべく多くの端末に対応できるようにする。

3 機能

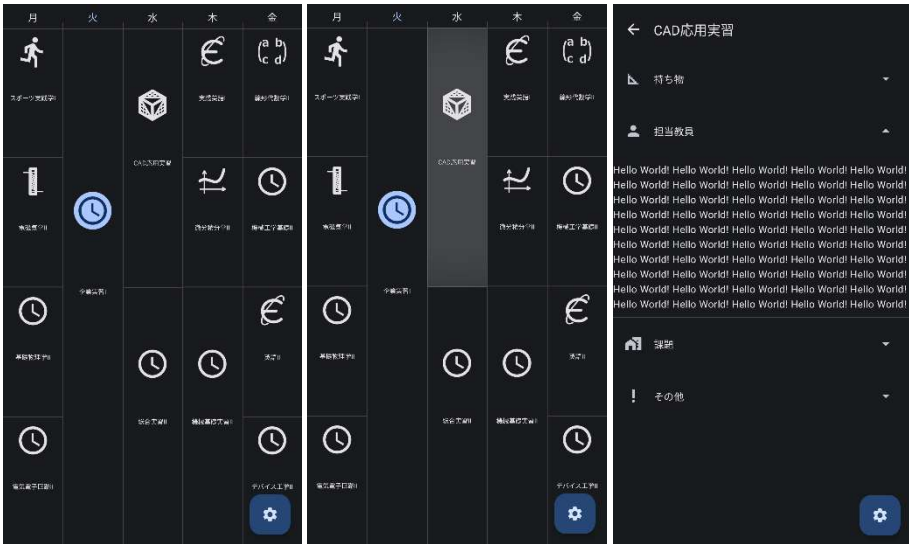
アプリケーションの実装手法の一部についてここでまとめていく。

3.1 ログイン画面



学年、学籍番号を入力してその後本人確認をする画面。本人確認には学生証を NFC リーダー部にタッチして行う。NFC に対応していない場合は QR コードを他の人のスマートフォンで表示してもらい、それを読み取る。下記の画像はその一部を切り取ったものである。

3.2 時間割



ログイン完了後、デフォルトでこの画面に飛ぶ。学年やコース、進路に応じてこの時間割で表示される内容が変わる。

また、今日が何曜日なのか、今何時間目なのかもこの画面からわかるようになっている。

マスをタッチすると授業の詳細がわかるようになっている。

4 実装

本項では、主にバックエンド側についてまとめていく。

まず、データの処理について。使用・要件に「学生情報など、後から変更される可能性のあるデータはプログラムとは別の場所に保存」と書いてあるように、保守性を向上させるために、別のファイルにまとめて書いてプログラム側でそのファイルからデータを取り出すように設計しなければならない。当初は XML ファイルで管理していたが、XML ファイルはテキストベースのファイルとなっており、プログラムと同じ要領で文法が読めないと変更・追加することができない。よって、Excel からデータを読み出すように変更した。Excel データの取り扱いの際に Apache POI という API を利用した。この API によって、VBA と同じような感覚で Java や Kotlin などを用いて Excel データの取り扱いが可能になる。

次に QR コードの処理について。QR コードについては ZXing を使用した。スマートフォンの画面が割れているなど、正常に表示できない可能性を考慮して、誤り訂正レベルを H にした。また、スマートフォンの画面やカメラの画素数の関係で、QR コードが正しく読み取れない問題を回避するため、学籍番号の情報に誤り訂正をかけた最小限の大きさに収まるように実装した。また、当初は学籍番号をそのまま QR コードに変換していたのだが、これでは一般的な QR コードジェネレータを使うだけで本人確認が通ってしまうため、本人確認の意味がなくなってしまう。この問題を解決するためにハッシュ関数を使用し、セキュリティを高めた。

最後にウェブメールを送受信できるようにするため、専攻科のホームページをアプリ内で表示する機能を実装することにした。現状、Jetpack Compose にはブラウジング機能を乗せるような関数が存在しない。そのため Android View 経由で Web View からウェブサイトを表示しなければならない。実装方法もいくつかあり、Google Chrome にリダイレクトする方法もあれば、完全にアプリケーション内で完結するように実装する方法もある。前者の方法は、Google Chrome で動かすため、ほとんどのウェブサイトが動作するのが大きな強みだ。しかし、そのアプリケーション内で完結しないため、あまり直感的ではない。反対に後者では、完全にアプリケーションで事が済むため、直感的ではあるが、一部動作しない箇所がある。現に、ウェブメールのページはアプリケーション内でのブラウザでは機能させることができなかった。

5 考察

本アプリケーションは現時点で 102MB もサイズがある。参考までに、近い容量のアプリケーションとしては YouTube がある (120MB 程度)。これは Excel のデータを未圧縮でしかもストレージ内に入れているのが問題だと考えられる。サーバにデータを入れておいて、数秒に一回読み込むみたいな処理にすればこれは大幅に削減できると考えられる。また授業変更の際に、リアルタイムで反映することもできるようになる。データの保存方法は今後も改善があると考えられる。

6 まとめ

本研究を通して、Android アプリケーションの開発について理解を深めることができた。また、保守性や再利用性の高いプログラムを作ることの重要性や難しさを体感することもできた。本研究で利用した、Jetpack Compose は正式リリースから 2 年ほどしか経っておらず、情報が少なかったり途中で大きな仕様変更があったりして、情報を集めるのに苦労した。情報の集め方に関しても本研究を通して成長できたと思う。問題点はまだまだたくさんあるので、後輩が研究を引き継いでアップデートを継続して行ってほしい。