

モバイルフレームワークを利用したモバイル端末用 加速器運転情報アプリケーションの開発

DEVELOPMENT OF ACCELERATOR OPERATION STATUS APPLICATION FOR MOBILE DEVICES USING A MOBILE FRAMEWORK

草野史郎 *^{A)}、佐藤政則 ^{B)}

Shiro Kusano*^{A)}, Masanori Satoh^{B)}

^{A)}Mitsubishi Electric System & Service Co., Ltd.

^{B)}High Energy Accelerator Research Organization (KEK)

Abstract

The KEK injector linac provides electron/positron beams to five different rings (SuperKEKB LER 4 GeV/HER 7 GeV/Positron Damping Ring 1.1 GeV, PF 2.5 GeV, PF-AR 6.5 GeV). For the long-term stable beam injection, it is very important to monitor the accelerator operation status. In recent years, the performance of mobile devices has been remarkably improved. By using these devices, it is easily possible to obtain the operation status of accelerators at anytime and from anywhere. Previously, we developed an application for displaying accelerator operation logs for iOS and Android OS platforms. However, since the different development environments were used for iOS and Android OS, it caused the poor maintainability of application. In this update, we use Flutter, a mobile framework for common development environments, to develop and update the accelerator operation log display application. This approach is helpful to improve the development and maintenance efficiency. In this paper, we report the details of this new application and future perspectives.

1. はじめに

KEK 電子陽電子入射器は、5つの異なるエネルギーのリング型加速器 (SuperKEKB LER 4 GeV, HER 7 GeV, positron damping ring 1.1 GeV, PF 2.5 GeV, PF-AR 5 GeV) にビームを供給している。長期間安定にビームを供給するためには、加速器運転状況を把握することが非常に重要である。近年、モバイル端末の進化は著しく、これらの端末を利用することでいつでもどこからでも加速器の状況を把握することができる。以前、iOS、Android OS で動作する加速器運転ログ表示用のアプリケーションの開発 [1] をおこなった。しかしながら、iOS および Android OS 用の開発言語が異なっていたため、保守および試験時間の増加など保守性に問題があった。今回、モバイルフレームワーク (共通開発環境) である Flutter [2] を用いて加速器運転ログ表示用のアプリケーションの更新をおこない、開発および保守の効率化を図った。本稿では、本アプリケーションについて、詳細を報告する。

2. アプリケーション概要

2.1 モバイルフレームワーク

モバイルフレームワークは、モバイル端末 (スマートフォンやタブレットなど) 上で動作するアプリケーションを開発する際に使用されるソフトウェアの基盤である。これらのフレームワークは、アプリケーションの開発プロセスを簡素化し、再利用可能なコンポーネントやツールを提供し、高品質かつ効率的なアプリケーションを作成することができる。モバイルフレームワークの特徴や機能を以下に示す。

- クロスプラットフォーム対応
複数のモバイルプラットフォーム (iOS、Android、Windows Phone など) で動作するアプリケーションを開発するためのクロスプラットフォーム対応を提供。これにより、1つのコードベースで複数のプラットフォーム向けのアプリケーションを作成することができる。
- UI コンポーネント
ユーザーインターフェース (UI) を構築するための事前定義されたコンポーネントが含まれており、ボタン、テキスト入力、リストビューなどの一般的な UI 要素を簡単に組み込むことができる。
- ネイティブ API へのアクセス
ネイティブ機能 (GPS、通知など) や API へのアクセスを提供。これにより、特定のプラットフォーム固有の機能にアクセスし、アプリケーションの機能を拡張することが可能である。
- 再利用可能なコード
ユーザーコミュニティで再利用可能なコンポーネントやライブラリが提供され、同じ機能を複数のアプリケーションで再作成する必要がなくなり、開発効率が向上する。

2.1.1 モバイルフレームワークの比較

モバイルフレームワークはさまざまな種類があり、以下には代表的ないくつかのモバイルフレームワークの言語、プラットフォーム、特徴を示す。

- React Native [3]
 - プログラム言語 JavaScript
 - プラットフォーム iOS, Android
 - 特徴
 - * コードの再利用が可能で、1つのコードベースで iOS と Android の両方に対応で

* skusa@post.kek.jp

きる。

- * ネイティブコンポーネントへのアクセスが容易である。
- * アクティブなコミュニティからのサポートが期待できる。
- * Facebook によって開発・メンテナンスされており、安定性が高い。

• Flutter

- プログラム言語 Dart [4]
- プラットフォーム iOS, Android, Web, Desktop
- 特徴
 - * ネイティブパフォーマンスに近い高速なアプリケーションを構築可能である。
 - * カスタム UI を作成するための豊富なウィジェットが用意されている。
 - * 1つのコードベースで複数のプラットフォームに対応している。
 - * Google によって開発・メンテナンスされており、アクティブなコミュニティが存在する。

• Xamarin [5]

- プログラム言語 C#
- プラットフォーム iOS, Android, macOS, Windows
- 特徴
 - * .NET フレームワークと統合しており、既存の C# のスキルを活用可能である。
 - * ネイティブ機能へのアクセスが可能である。
 - * Microsoft によって開発・メンテナンスされており、豊富なドキュメンテーションが提供されている。

2.1.2 モバイルフレームワークの選定

本アプリのモバイルフレームワークとして、Flutter を採用した。Flutter は、ほかのモバイルフレームワークと比べて高度に最適化された Dart 言語によって動作しているため、ネイティブアプリケーションに近い動作速度をもっている。UI は、Google 推奨のマテリアルデザインを採用しており、豊富なウィジェットライブラリが提供されている。これにより、独自のカスタム UI を簡単に作成ができ、アプリケーションの外観を柔軟にカスタマイズすることができる。さらに、ホットリロード機能が備わっているため、コードの変更をリアルタイムに反映させながら、素早く開発を進めることができる。Google がバックアップする Flutter の開発コミュニティは非常に活発であり、多くの情報源やサードパーティのパッケージが利用可能なことから、将来性が高いと判断して本アプリのモバイルフレームワークとして採用した。

2.2 開発環境

アプリ開発開発環境は、Windows PC 上でおこなった。ただし、iOS 用アプリに関しては、iOS 用にコンパイルする macOS が必要となるため、コンパイル環境用として MacBook Pro を採用した。開発環境用 Windows PC

と Mac PC のスペックを Table 1 に示す。

Table 1: Development Environment

PC	MPro-T330	MacBook Pro
CPU	R3.7GHz Intel Core i9	R2.3 GHz Intel Core i5
Memory	64GB	16 GB
OS	windows11	macOS Big Sur

2.3 Android Studio

本アプリの開発は、Android Studio [6] でおこなった。Android Studio は、Google が提供する Android OS 向けアプリ開発用の IDE である。Android Studio は、Android 開発者向けのサイト Android Developers から無償でダウンロードすることができる。Android Studio は開発するための OS に制限がなく、Windows, macOS, Linux および Google Chrome OS に対応している。Android Studio は、Flutter のプロジェクト開発においても、コードの自動補完やリファクタリング、デバッグ、テスト、エミュレーションなどの機能を効果的に利用できる。Figure 1 に Android Studio の表示画面を示す。

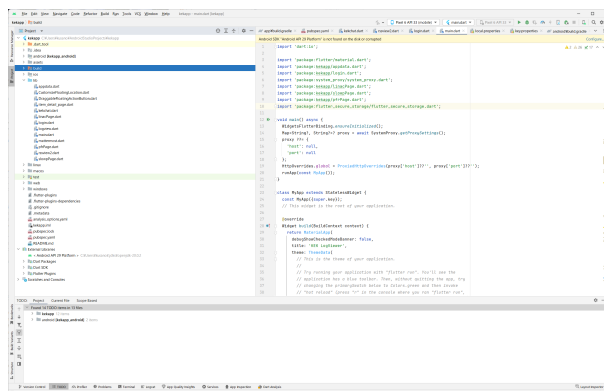
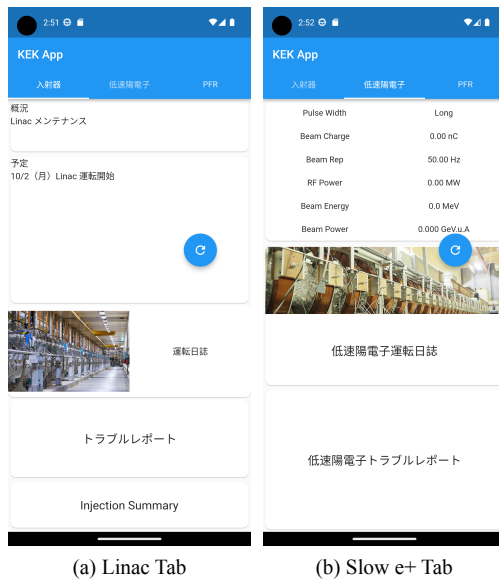


Figure 1: Screen shot image of Android Studio.

3. 加速器運転情報アプリ

Figure 2 にアプリのトップ画面を示す。(a) が入射器、(b) が低速陽電子のタブページある。入射器のタブページにおいては、加速器施設の概況および予定を表示している。運転日誌、トラブル、Injection Summary は、以下に記述する各パネルへのリンクボタンとなっている。低速陽電子のタブページでは、現在の Pulse Width, Beam Charge, Beam Rep, RF Power, Beam Power の値を表示している。運転日誌、トラブルは、以下に記述する各パネルへのリンクボタンとなっている。両タブページの概況および予定、各パラメータの値は、画面中央にある Floating Button を押すことで更新される。



(a) Linac Tab

(b) Slow e+ Tab

Figure 2: Screen image of accelerator operation status application.



(a) Linac Operation Log

(b) Slow e+ Operation Log

Figure 3: Screen shot of operation log.

3.1 運転日誌パネル

Figure 3 に入射器および低速陽電子の運転日誌パネルを示す。運転日誌パネルは、すでに所外向けに提供している電子ログブックシステムの Web ページを利用している [7]。画面下部に表示されている前ボタンは前シフト、後ボタンは次シフト、ホームボタンはトップ画面に遷移する。電子ログブックシステムの Web ページでは、関係者以外が閲覧できないようにユーザー認証で管理されている。本アプリは、接続する度パスワードを入力する手間を省くため、flutter_secure_storage モジュールを用いて端末本体にユーザーおよびパスワードを暗号化して保存している。

3.2 トラブルパネル

Figure 4 にトラブルパネルの表示画面を示す。トラブルの表示は、入射器で提供している RSS フィードの Web ページを利用して表示している。パネルでは、直近のトラブル件数 10 件をリスト表示している。各リストをタップするとトラブルの詳細が表示される。

3.3 Injection Summary パネル

Figure 5 に入射器の Injection Summary の表示画面を示す。Injection Summary は、入射器で提供している Injection Summary の Web ページを利用して表示している。パネルでは、SuperKEKB HER、SuperKEKB LER、PF、PF-AR のそれぞれの蓄積電流、ビーム繰り返し、入射効率、BT End での電荷量をグラフで表示している。グラフの横軸は時間軸となっており、ボタンで 24H、8H、2H を選択可能となっている。また、各グラフをタップするとグラフが拡大して表示される。

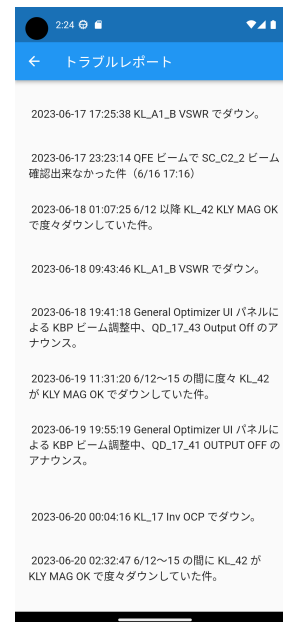


Figure 4: Screen shot image of trouble report.

4. 課題

4.1 通知機能および加速器情報の記録

以前のアプリでは、Google 社が提供する mobile Backend as a Service (mBaaS) である Firebase を利用した。アプリがバックグラウンドにある場合でも利用できるプッシュ通知機能の Firebase Cloud Messaging やユーザー認証機能の Firebase Authentication、リアルタイムデータベースの Firebase Realtime Database などのサービスを用いて通知機能や加速器情報の記録用データベースを実装した。本アプリにおいては、機密情報管理の観点から外部サービスを利用することは好ましくない

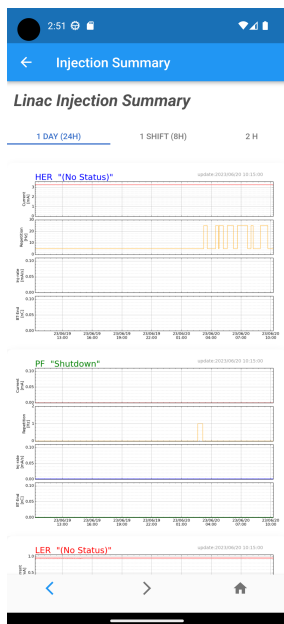


Figure 5: Screen shot image of linac injection summary.

と判断し、KEKの所外向けWebサーバーに情報を記録することにした。その記録方法は、加速器の現在の情報をXMLファイルに記録し、FlutterのXMLパッケージによりXMLファイルに記録されているタグ情報を元にそれぞれの加速器の情報を表示した。プッシュ通知機能においては、mattermostサーバーを利用することを検討している。

4.2 アプリの配布方法

以前のアプリの配布方法は、iOS用では、AppStore (TestFlight) による配布、Android用アプリは、任意のWebサイトから配布でおこなっていた。今回のアプリでは、Firebase App Distributionを利用した配布を検討している。Firebase App Distributionは、アプリ利用者がアプリ作成者の招待状により、Firebase App Distributionアプリのインストールをおこなう。Firebase App Distributionアプリを通じてアプリのインストールができる。Firebase App Distributionのメリットは、iOSかAndroidかを問わず、開発途中のアプリを配布することが可能であり、アプリを更新した際、Firebase App Distributionアプリによりアップグレード通知のサービスを受けることができる。

5. まとめ

モバイルフレームワークを利用したモバイル端末用加速器運転情報アプリケーションの開発をおこなった。現在、2~3名のユーザーで試験運用をおこなっている。以前のアプリと比べて、1つのコードベースで複数のプラットフォーム (iOS、Android、Web、Desktop) に対応しているため、開発およびテストやデバッグの時間が短縮され、開発効率が向上した。今後は、さらに多くの関係者が利用できるような本格的な運用を目指す予定である。

参考文献

- [1] S. Kusano *et al.*, “モバイル端末を利用した加速器運転情報アプリケーションの開発”, Proceedings of the 19th Annual Meeting of Particle Accelerator Society of Japan, October 18-21, 2022, Online (Kyushu University).
- [2] <https://flutter.dev/>
- [3] <https://reactnative.dev/>
- [4] <https://dart.dev/>
- [5] <https://docs.microsoft.com/ja-jp/xamarin/>
- [6] <https://developer.android.com/studio>
- [7] T. Kudou *et al.*, “入射器における電子ログブックシステムの更新”, Proceedings of the 8th Annual Meeting of Particle Accelerator Society of Japan, August 1-3, 2011, Tsukuba, Japan.