

農家向け作物栽培支援アプリにおける オフライン対応のテスト手法の検討

重本 拓己[†] 東野 正幸[‡] 川村 尚生[§] 坪 充[¶]
鳥取大学[†] 鳥取大学[‡] 鳥取大学[§] 鳥取大学[¶]

1 はじめに

降雨量の限られた乾燥地で農業従事者が安定かつ自立した作物栽培を行うには、降雨量に基づいた作物収量の予測が重要である。乾燥地では降雨量を記録するためのインフラ整備が難しい地域が多い。しかし、近年ではスマートフォンとインターネットが普及しつつある。このため、農業従事者がスマートフォンを用いて降雨量等を記録し、これらのデータをインターネットを介して集積・分析することで、当該地域での作物収量を予測できる可能性がある。

そこで我々は、農業従事者が手作りの雨量計を用いて測定した圃場の降雨量と栽培する作物に関するデータをモバイルアプリケーション（以下、本アプリ）によりクラウドへ集積し、集積したデータと気候予測データを用いた分析により得られた作物収量の予測結果を元に、農業従事者に播種や施肥の適切な量や時期を提示するアプリを開発している [1]。

しかし、都市部から離れた圃場では通信インフラが整備されていない場合が多い。本アプリはこれまでクラウドを用いたオンライン環境での利用を前提としていたため、通信インフラが整備されていない圃場では降雨量等の記録を行うことができず、利便性に課題があった。そこで本アプリを利用する際にモバイル端末がネットワークに接続されているかどうかに関わらず、入力データをローカルデータベースに保存し、ネットワークに接続された時にローカルデータベースとクラウドデータベースをバックグラウンドで自動的に同期する手法を用いたオフライン対応を提案した [2]。

しかし、オフライン対応のために実装した機能のテストを手動でしか行っておらず、テストは自動化されることが望ましい。そこで本稿では、開発したオフライン対応の機能を示し、その自動テスト手法を検討する。

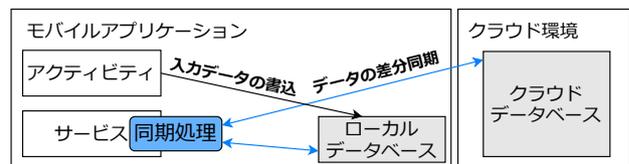


図1 ローカルデータベースへの書き込み動作とクラウドデータベースとのバックグラウンド同期動作

2 オフライン対応の設計と実装

2.1 ローカルデータベースへの保存

本アプリにローカルデータベースを用意し、デバイスのネットワーク接続状態に関わらず、入力データはローカルデータベースに書き込む（図1）。入力データのローカルデータベースへの保存は、本アプリで使用しているクラウド型のデータベースである Firebase Cloud Firestore（以下、Firestore）のオフラインサポート機能^{*1}を使用して実現している。

2.2 バックグラウンド同期とスケジュール

本アプリは Android アプリケーションとして実装されている。Android アプリケーションは基本的にアクティビティと呼ばれるコンポーネント単位で動作するが、アクティビティはユーザによる操作だけでなく Android 端末の状況に応じてアクティビティのライフサイクル^{*2}の様々な状態に自動的に遷移する。そこで、ローカルデータベースとクラウドデータベースをバックグラウンドで確実に同期するために、バックグラウンド処理に特化したコンポーネントであるサービスを用意し、ローカルデータベースとクラウドデータベースを同期するタスクを持たせる。このタスクが定期的に実行されるようにスケジュールを設定することで、アクティビティの状態に依存せず、定期的にローカルデータベースとクラウドデータベースがバックグラウンドで同期される（図1）。

Firestore のオフラインサポート機能により、オフライン状態でのローカルデータベースに対する変更がデバイス

A testing method for an offline-first feature of a cultivation support app for farmers

[†] Takumi Shigemoto, Tottori University

[‡] Masayuki Higashino, Tottori University

[§] Takao Kawamura, Tottori University

[¶] Mitsuru Tsubo, Tottori University

^{*1} Firebase Cloud Firestore - Access data offline, <https://firebase.google.com/docs/firestore/manage-data/enable-offline>

^{*2} The Activity Lifecycle - Android Developers, <https://developer.android.com/guide/components/activities/activity-lifecycle>

がオンラインに戻る時にクラウドデータベースに同期される。しかし、Firestore はアクティビティのライフサイクルに従って動作するため、アクティビティの状態が *Activity running* の状態でしか同期が行えない。このため、オフライン状態でデータの記録後に本アプリを操作しない限り、クラウドへのデータ集積が滞る問題がある。この問題をサービスを用いて解決している。

サービスに持たせる同期タスクのスケジューリングは、サービスを扱いやすくするライブラリである *WorkManager*^{*3} を使用して実現している。同期タスクとしては、データベースに新しくデータが書き込まれたことをトリガとして未同期のデータを同期する Firestore の挙動を利用して、ダミーのデータをクラウドデータベースに書き込む処理を行なっている。

3 自動テスト手法の検討

3.1 自動テストの設計

本アプリのオフライン対応は、デバイスがオフラインの場合でもローカルデータベースへデータの記録を可能であり、デバイスがオンラインに戻った時に、データが自動的にクラウドデータベースに同期される。つまり、以下の2点が行えることを確認する必要がある。

- デバイスがオフライン時に本アプリに入力されたデータのローカルデータベースへの保存
- デバイスがオンラインに戻った時の、オフライン時に入力されたデータのバックグラウンドでのクラウドデータベースへの同期

そこで、まずデバイスがオフラインの状態ですべてのアプリを用いてデータを記録する。その後、デバイスをネットワークに接続し、オフライン時に入力したデータがクラウドデータベースに同期されることを確認する。ただし、アクティビティの状態が *Activity running* の状態（フォアグラウンド）であっても、それ以外の状態（バックグラウンド）であっても、同期される必要がある。そこで、自動テストにおいてアクティビティを *Activity running* の状態（フォアグラウンド）とそれ以外の状態（バックグラウンド）へ明示的に変化させることで、フォアグラウンドとバックグラウンドのどちらでも正しくクラウドデータベースに同期されることを確認する手法を取る。

3.2 自動テストの実装

自動テストの実装には、本アプリの利用手順に沿ったシナリオテストの作成、デバイスのネットワーク接続状態の切替機能の開発、Android のアクティビティの状態の変更機能の開発が必要となる。

デバイスのネットワーク接続状態の切替機能には、UI テストフレームワークである *UI Automator*^{*4} を使用する。UI Automator の *executeShellCommand* メソッドを用いることで、ルート化されたデバイス上で *adb* シェルコマンドを実行することができる。この機能により、デバイスの Wi-Fi の有効または無効を切り替える *svc wifi enable/disable* コマンドと、セルラー回線接続の有効または無効を切り替える *svc data enable/disable* コマンドを使用し、デバイスのネットワーク接続状態を切り替える。また、Android のアクティビティの状態の設定機能についても同様に、UI Automator の *pressHome* メソッドを用いることで、アクティビティの状態を *onStop()* の状態（バックグラウンド）に変更する。また、ローカルデータベースとクラウドデータベースの同期確認には、Firestore の *waitForPendingWrite* メソッドを用いることでローカルデータが全てクラウドへ書き込まれているかどうかを確認できる。

3.3 自動テストの動作実験

実装した自動テストにより、本オフライン対応の妥当性を確認する。実験環境として、デバイスには Android Emulator 上で作成した Pixel 5 の Android 11 版をルート化したものを使用する。クラウドデータベースは Firebase で構築した Firestore を使用する。実験の結果、アクティビティの状態が *Activity running* 以外の状態でもオフライン時に記録されたデータがクラウドデータベースに正しく同期されることが確認できた。このことから、本アプリのオフライン対応のために実装した機能が正常に動作していることが確認できる。

4 おわりに

本稿では、本アプリのオフライン対応のために実装した機能を示し、デバイスのネットワークの接続状態と Android のアクティビティの状態を変化させることでオフライン対応の機能の妥当性を確認する自動テスト手法の実装及び動作実験を行った。実験の結果より、本オフライン対応がアクティビティの状態によらず、正常に動作することが確認できた。

参考文献

- [1] 元田匡哉, 東野正幸, 川村尚生, 近藤克哉, 坪 充: 乾燥地における作物生育の安定化に向けた農家による降雨量データの集積と作物収量予測のためのモバイルアプリケーションの検討, 情報処理学会研究報告, Vol. 2020-MBL-97, No. 1, pp. 1-5 (2020).
- [2] 重本拓己, 東野正幸, 川村尚生, 坪 充: 作物収量予測のためのモバイルアプリケーションのオフライン対応, 2022 年度 (第 73 回) 電気・情報関連学会中国支部連合大会講演論文集 (2022).

^{*3} App Architecture: Data Layer - Schedule Task with WorkManager - Android Developers, <https://developer.android.com/topic/libraries/architecture/workmanager>

^{*4} UI Automator - Android Developers, <https://developer.android.com/training/testing/ui-automator>