

一個使用 Flutter 開發、基於整潔架構的點名提醒系統

An auto roll-call reminder system written in Flutter Clean Architecture

專題編號：111-CSIE-S020

執行期限：110 年第 1 學期至 111 年第 1 學期

指導教授：鄭有進

專題參與人員：108820003 林天佑

一、摘要

在疫情的衝擊下，校園裡有越來越多課程都開始採用數位點名系統來進行線上點名，但我們發現，有時候會出現老師已經開始進行點名，學生也有到課，卻仍出現學生不知正在進行點名的狀況發生。

本專題的目的是在現有的校園 APP 上，新增一個針對正在進行中的線上點名活動的提醒功能，當老師發起一個進行中的點名活動時，使用者(學生)能及時地從行動裝置上收到點名通知，且可直接透過用戶端來達成點名操作，藉此提高點名的效率。

二、緣由與目的

線上點名系統能提供老師在線上開啟一個限時的點名活動，學生必須要在這個時限之內進入該系統進行操作(例如：點擊畫面上的某按鈕)，以完成簽到流程。如果超時簽到，則可能會被標記為遲到，或是直接記錄為曠課。

以本校為例，有很多老師採用了 Zuvio 這套系統來進行點名，然而我們卻經常發現有學生直到點名活動都結束了才注意到不久之前有開放點名，尤其是在遠距上課的情況下，學生能夠聽到老師說即將開放或正在開放點名的可能性就降低了不少，舉例來說，線上課程可能發生以下幾種狀況：

1. 學生因為突發狀況而無法即時加入線上課程的視訊會議，導致無法得知已經開始點名。

2. 老師僅透過文字等較不容易察覺的方式通知學生即將開始點名，但是學生注意力都放在視訊畫面上，難免會沒注意到一旁的文字訊息，導致無法得知已經開始點名。

TAT 是一個專門為本校師生開發的校園 APP，支援 Android & iOS 雙平台且開源。只要持有本校有效的校園帳號，就可以透過 TAT 來查課表、查成績、看行事曆、看公告等等。

而 TAT 自兩年前發布至今已經普遍地被本校學生使用，雙平台的活躍使用者數量也幾乎累計達到 6000 多人，因此，我們將利用這個優勢，在 TAT 上新增一個點名提醒的通知功能，學生可以針對 Zuvio 課程設定一個監控排程，當 Zuvio 平台於該排程之時段開始進行點名活動時，TAT 就會及時地發出通知，藉此提醒學生點名已經開始，要盡快簽到。

三、流程與策略

● 流程

主要採用 Kanban 的方式進行開發，並使用 GitHub project (new) 的功能來管理 issue & pull requests。

● 策略(用戶端)

TAT 的開發從一開始就面臨著許多挑戰。要在 TAT 上開發一個新的功能，首先必須解決現有程式碼的一些重大問題，並進行大規模的重構。

我們將開發的方向分成三個不同的路線：

1. New Features
2. Flutter Null-Safety Migration
3. NTUT Connection Bugs resolve

而這幾個開發路線主要緣起於以下三項因素：

■ 框架與依賴套件升級

Flutter [1] 是一個由 Google 發起的跨平台開源應用程式框架，為 TAT 的主要開發技術，因此，我們得以維護一份程式碼，卻能發布到多個不同的平台上。

而近期 Google 在其年度開發者大會 Google I/O 2022 [2] 上正式發佈了下一代的 Flutter 主要版本：Flutter 3。

為了獲得新版本帶來的好處，以及達到重構現有程式碼的目標，我們決定將 TAT 使用的 Flutter 版本從 1 升級到 3。

此外，升級的過程也包含自 Flutter 2 發布以來的 Null Safety (空值安全) [3] 升級，它可以使程式碼嚴格地區分出「可能為 null」以及「不可能為 null」的型態，大幅地降低意外存取空值的風險，還能提高可讀性、效能、安全性等等。

■ 現有程式碼混亂

該 APP 起初並非由我們開發，且原先的程式碼存在著以下的阻礙必須解決：

- 缺乏整潔的架構、高耦合低內聚
- 變數命名難以理解其意圖
- 程式區塊含糊不清
- 濫用動態型別
- 缺乏註解、文件，與單元測試、

整合測試、UI 測試等等

- 無考量到效能與安全性等等的議題
- 有些套件已經停止支援，需要遷移至新版本或改用其他套件

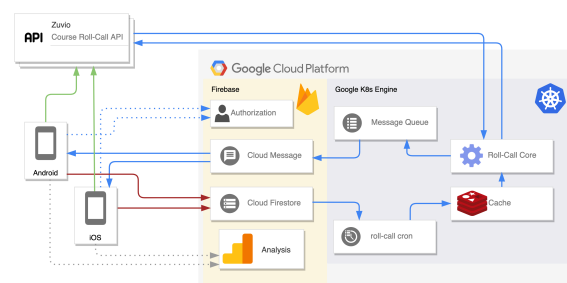
因此我們為了更有效率的解決以上問題，大多情況下將優先考慮直接搭配 Null-Safety Migration 的步驟，一起重構所有範圍內的功能與邏輯。

■ 校方 API 不穩定

TAT 並非由本校官方出品，目前也未與學校簽訂任何形式的契約，因此每當校方後台 API 有變動時，TAT 就會直接面臨各種無預警的連線錯誤，且校方現有的 API 也缺乏統一、老舊、充滿各式各樣未遵守 WEB RFC 的資料，導致我們必須在開發新功能的同时也要解決現有的連線問題。

四、架構流程

本專題分為用戶端(TAT)應用程式功能，與後端系統大部分。



(後端系統架構簡圖)

後端系統主要分為兩區域：Firebase 與 GKE Cluster。其中 Firebase 作為與用戶端溝通的主要介面，而 GKE 中部署的服務將主要負責與外部 API 進行互動，以 Golang 或 Rust 進行開發。

目前計畫的運作模式如下：

1. 使用者在用戶端設定數個點名監看排程，並將這些設定值儲存到 Firestore [4] 中，而此過程我們將使用 Firebase Auth 來進行身份驗證與 Firestore 權限控管。

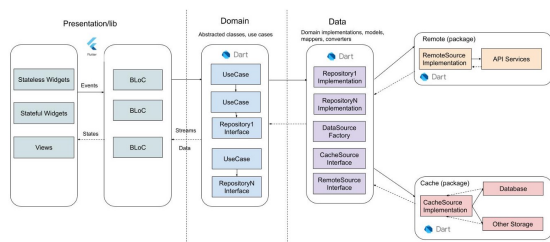
2. 每個上課日開始之前的某個時間點，cron job service 會從 Firestore 提取最新的排程設定，將其整理成工作清單後再儲存至 cache service 中。

3. cron job service 在上課期間利用上個步驟整理好的工作清單，定時 trigger core service，使其發送請求至 Zuvio API。

4. core service 得到 Zuvio API 的答覆後，會將資訊送入 Message Queue service。

5. Message Queue service 再將包裝好的訊息交給 Firebase Cloud Message (FCM) [5]，使其發送通知到用戶端。

除此之外，我們使用 Google Analysis service 來彙整後端各個行為的 log，以及用戶端使用過程所產生的 log。



(TAT 設計架構簡圖)

用戶端的所有新功能與使用者介面將在 Dart Sound Null Safety 的條件下以 Bloc Pattern [6] Based Clean architecture 進行設計，並使用 Riverpod [7] 作為 Dependencies Injection tool。

而該功能會新增 3 個頁面：

■ Zuvio 登入頁面



(登入頁面預覽圖)

引導使用者登入 Zuvio 平台，並儲存其帳號資訊於裝置安全區域。

■ 新增點名監看排程頁面



(新增點名監看排程牌卡預覽圖)

使用者可於該頁面檢視已登入之 Zuvio 帳號下有哪些課程，並從各個課程中選取星期與時段，新增點名監看排程。

■ 點名監看管理頁面

該頁面將顯示目前已排程之監看服務，並可於該頁面移除或暫停監看排程。

參考資料

- [1] <https://flutter.dev/>
- [2] <https://io.google/2022/products/flutter/>
- [3] <https://dart.dev/null-safety>
- [4] <https://firebase.google.com/docs/firestore>
- [5] <https://firebase.google.com/docs/cloud-messaging>
- [6] <https://bloclibrary.dev>
- [7] <https://riverpod.dev/>