112學年第一學期 APP程式設計

**期末專案成果報告**

題目：Music Player APP

組員：C110151138 黃吉菘

日期：113年1月5日

1. 動機與目的

動機：

音樂愛好者的需求：許多人喜歡隨時隨地聆聽他們喜愛的音樂，透過一個優秀的音樂播放器，可以更好地滿足這種需求。

優化用戶體驗：現有的音樂播放器可能存在使用上的不便，透過自己開發的應用程式，可以根據用戶反饋來優化使用者介面，提供更直觀且個人化的體驗。

學習Flutter技術： Flutter是一個現代化的跨平台框架，透過這次開發，可以學習到如何使用Flutter來建立具有吸引力和高效性能的應用程式。

開發自己的品牌：透過開發一個專屬的音樂播放器，不僅可以提供優秀的服務，同時也能建立自己的品牌形象，擁有獨特的特色。

目的：

提供優質音樂體驗：開發一個功能豐富、用戶友好的音樂播放器，以提供優質的音樂聆聽體驗，包括高音質、播放清單管理、智能推薦等功能。

個人化推薦：透過應用程式學習用戶的喜好，提供個性化的音樂推薦，使用戶能夠發現新的音樂並保持對應用程式的興趣。

跨平台支援：利用Flutter的跨平台特性，確保應用程式可以在不同的移動平台上運行，滿足更廣泛的用戶需求。

開發社群參與：創建一個開源項目，吸引開發者參與，共同改進應用程式，形成一個繁榮的社群。

不斷優化與更新：定期更新應用程式，加入新功能、修復問題，確保用戶擁有最新且高品質的使用體驗。

2. 背景說明

　　音樂在現代生活中扮演著重要的角色，人們愈發依賴手機和移動設備來聆聽他們喜愛的音樂。在這樣的背景下，開發一個使用Flutter框架製作的音樂播放器應用程式是相當有意義的。

隨著科技的不斷進步，跨平台開發成為一個重要的趨勢。Flutter作為一個由Google開發的開源UI工具包，具有高度的靈活性和效能，使得開發者能夠同時在Android和iOS平台上構建出色的應用程式。這意味著使用Flutter開發音樂播放器可以確保用戶在不同設備上都能享受到統一且流暢的使用體驗。

此外，音樂播放器應用程式的開發旨在滿足現代用戶的需求，包括但不限於高音質播放、智能推薦、個人化歌單管理等功能。透過使用Flutter，開發者能夠更輕鬆地實現這些複雜的功能，同時確保應用程式的性能和反應速度達到最佳水平。

開發一個音樂播放器應用程式也提供了一個學習Flutter的機會。Flutter的編程模型和豐富的組件庫使得開發過程更加直觀和高效。這將為開發者提供在跨平台應用程式開發領域取得更多技能和經驗的機會。

3. 系統架構

架構圖說明

1.音樂檔案：用於播放音樂的檔案。

2.圖片檔案：用於顯示圖像的檔案。

3.圖示：用於代表應用程式的圖形。

4.標題：顯示應用程式名稱的文字。

5.進度條:顯示音樂進度

6.按鈕：用於讓使用者與應用程式互動。

4. 操作流程

流程圖說明

1.登入畫面：

用戶打開APP後，首先進入主畫面。

2. 點擊播放按鈕：

成功登入，用戶點擊播放按鈕即可播放音樂。

用戶在界面上找到輸入金額的區域，輸入欲轉換的金額。

3. 拖動進度條：

用戶拖動進度條即可選擇音樂播放的秒數。

4. 快轉10秒：

用戶按下快進10秒的按鈕後，音樂會跳轉至10秒後。

5. 倒放10秒：

用戶按下倒放10秒的按鈕後，音樂會跳轉至10秒前。

5. 成果說明

1.音樂播放功能：

使用audioplayers套件實現了基本的音樂播放功能，可以播放指定的音樂檔案（在此例中為 "audio/idol.mp3"）。

透過初始化音樂播放器、監聽播放進度和狀態改變等事件，實現了對音樂的控制和同步。

2.使用者介面設計：

以Flutter的Widget為基礎，設計了一個簡潔的使用者介面。

介面包括歌曲封面、歌曲標題、歌手等資訊，以及音樂播放控制元件（播放/暫停按鈕、進度條、快進/快退按鈕）。

3.進度條和時間顯示：

使用Slider元件實現了音樂播放進度的顯示和操作功能，並顯示了當前播放時間和總播放時間。

透過對進度條的操作，用戶可以自由調整音樂的播放進度。

4.快進/快退功能：

透過IconButton元件實現了快進和快退功能，使用戶可以在歌曲中進行細微的進度調整。

5.格式化時間顯示：

使用formatDuration函數將播放時間格式化為易讀的形式，方便用戶了解當前播放進度和總播放時間。

6. 結論

總體而言，這個Flutter音樂播放器應用程式提供了基本的音樂播放功能，同時設計了直觀且易於使用的使用者介面，使得用戶可以輕鬆地控制和享受音樂播放體驗。

7. 建議（類似專題之改進方向）

這個音樂播放器應用程式已經有了一個堅實的基礎，但還有一些改進和擴展的方向可以考慮，以提高其功能性和用戶體驗。

1.播放清單功能：

添加支援播放清單的功能，讓用戶可以創建、管理和播放自己的音樂清單。

2.歌詞顯示：

整合歌詞顯示功能，使用戶能夠同步閱讀歌曲的歌詞。

3.主題和個性化：

提供不同主題和個性化選項，讓用戶可以根據自己的喜好自訂應用程式的外觀和風格。

4.後台播放支援：

實現後台播放功能，使得用戶可以在應用程式切換到後台時繼續播放音樂。

5.音效增強：

整合音效增強功能，例如均衡器或音效調整，以提供更豐富的音樂聆聽體驗。

6.社交分享：

添加社交分享功能，讓用戶能夠分享他們正在聆聽的歌曲到社交媒體平台。

7.搜尋和篩選：

實現搜尋功能，讓用戶可以輕鬆找到並播放他們喜愛的歌曲。同時，添加篩選和排序選項，以提高用戶對音樂的管理效率。

8.錯誤處理和提示：

加強錯誤處理機制，提供用戶友好的錯誤提示，以提高應用程式的穩定性。

9.跨平台同步：

若可能，考慮添加用戶資料和播放清單的跨平台同步功能，讓用戶在不同設備上能夠保持一致的音樂體驗。

8. 參考文獻

1. Flutter 3 + Audioplayers 实现手机音乐播放器 Android & iOS 移动开发

<https://youtu.be/0lY1g4sm_2w?si=YIwXJCriz7wXgP1t>

9. 檔案連結

<https://github.com/SmallSong0302/C110151138_final>

10. 附錄

(組員人頭簡介及分工情形)

|  |  |
| --- | --- |
| 組員 | 介紹 |
| 黃吉菘 | 程式、報告 |