



POKE-DEX

組別一

110820031 翁廷豪

110820048 許家睿

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

Poke Dex

來自死忠寶可夢粉絲的APP

寶可夢圖鑑 每日寶可夢 寶可夢挑戰 個人紀錄



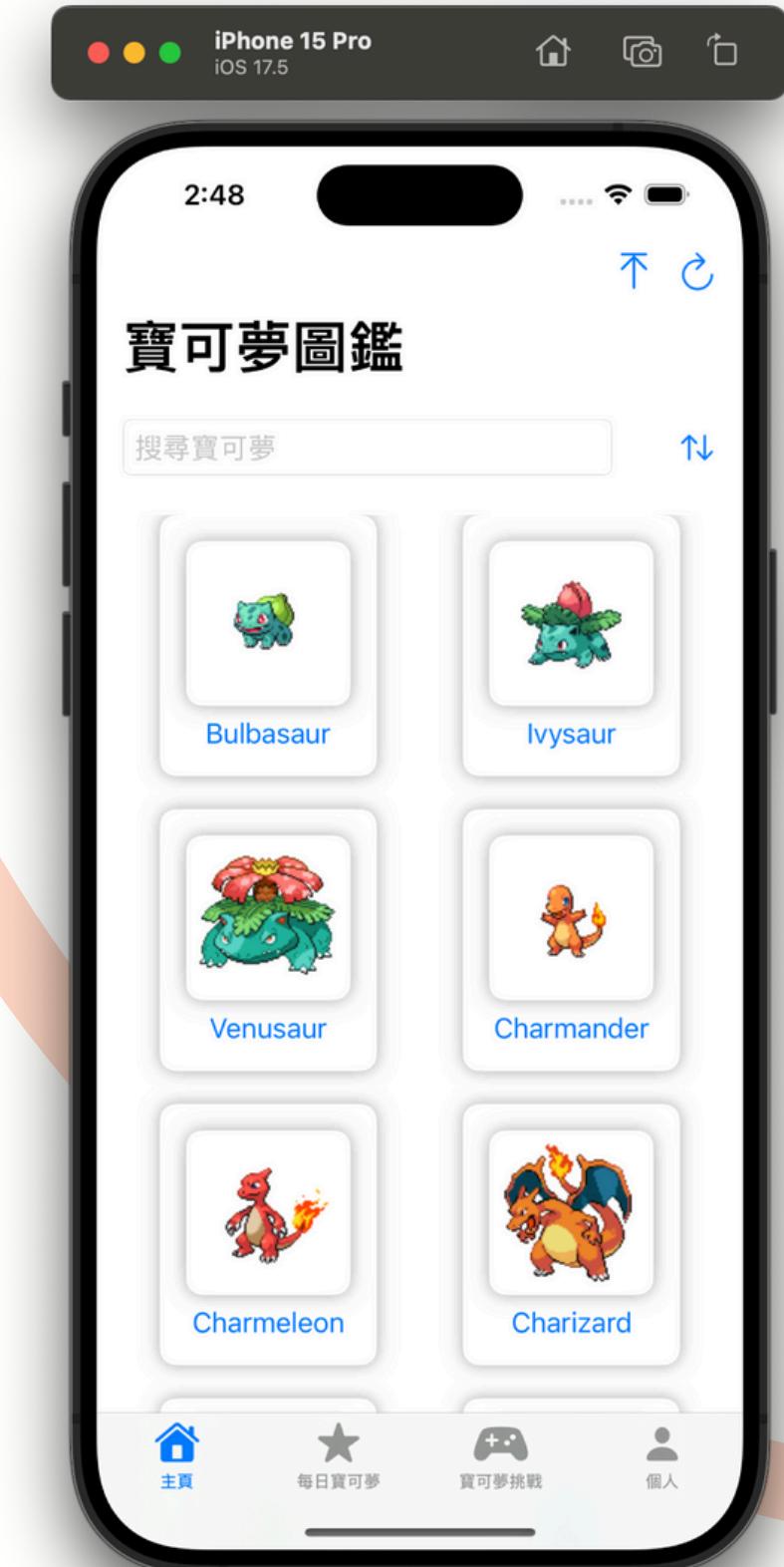
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

APP FEATURE

寶可夢圖鑑

每日寶可夢 寶可夢挑戰 個人紀錄

Displays a grid of Pokémon fetched from the PokeAPI. Users can search for specific Pokémon and sort them by various criteria such as ID, weight, and height.



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

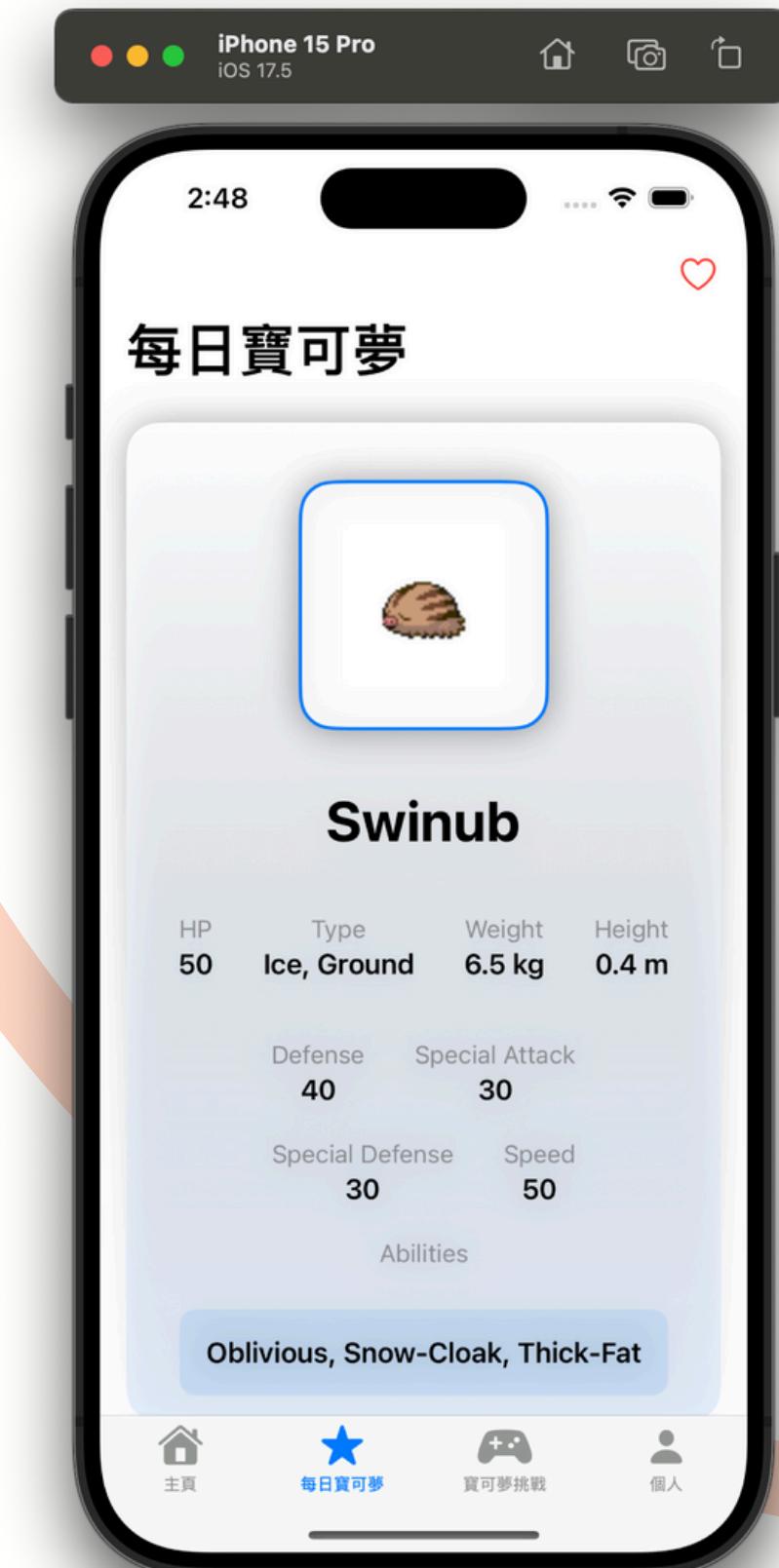
APP FEATURE

寶可夢圖鑑

每日寶可夢

寶可夢挑戰 個人紀錄

Recommends a random Pokémon each day, displaying detailed information about it.



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

APP FEATURE

寶可夢圖鑑 每日寶可夢 寶可夢挑戰 個人紀錄

Provides a quiz game where users guess which Pokémon is taller or heavier based on provided options.

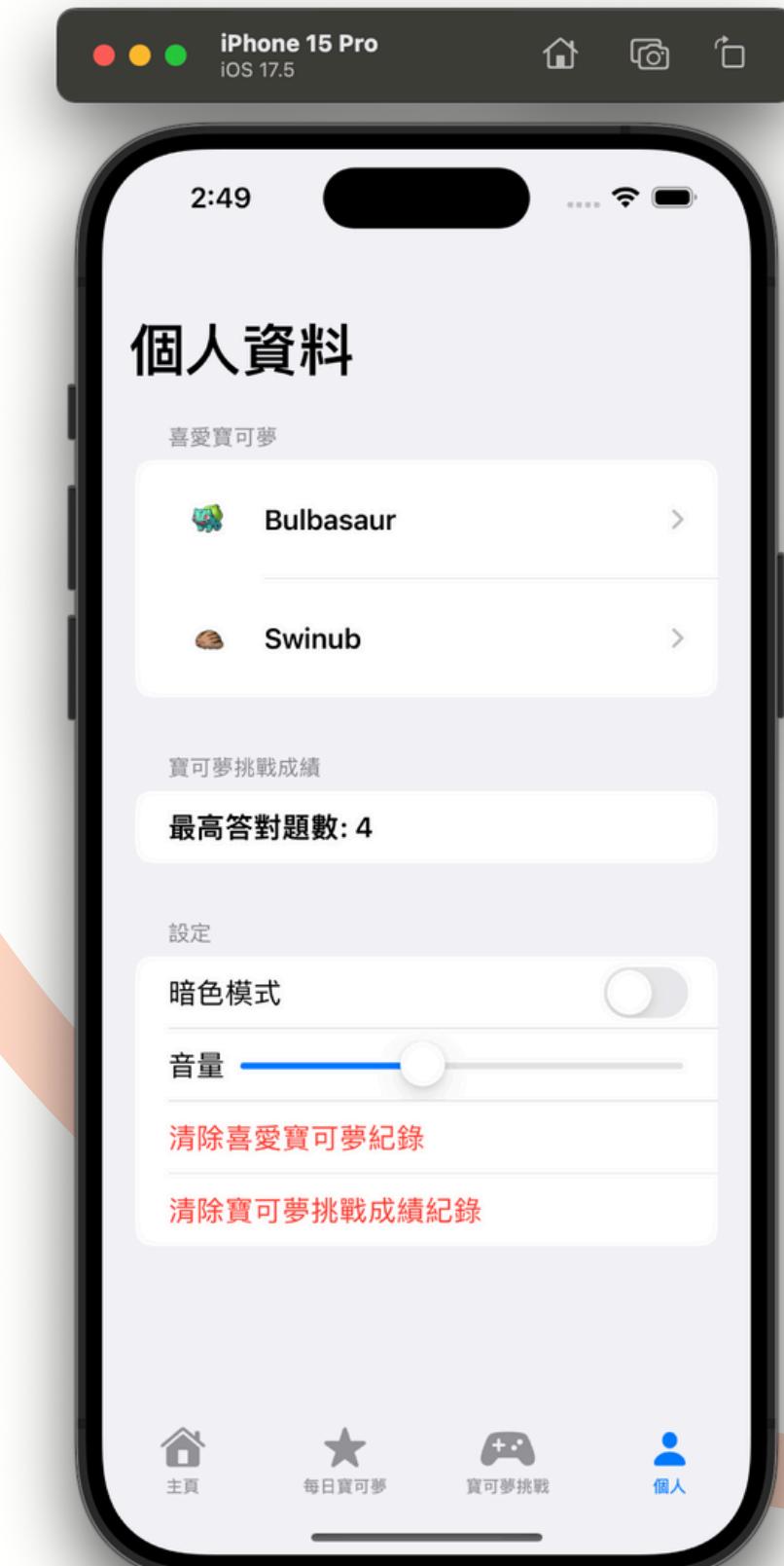


- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

APP FEATURE

寶可夢圖鑑 每日寶可夢 寶可夢挑戰 個人紀錄

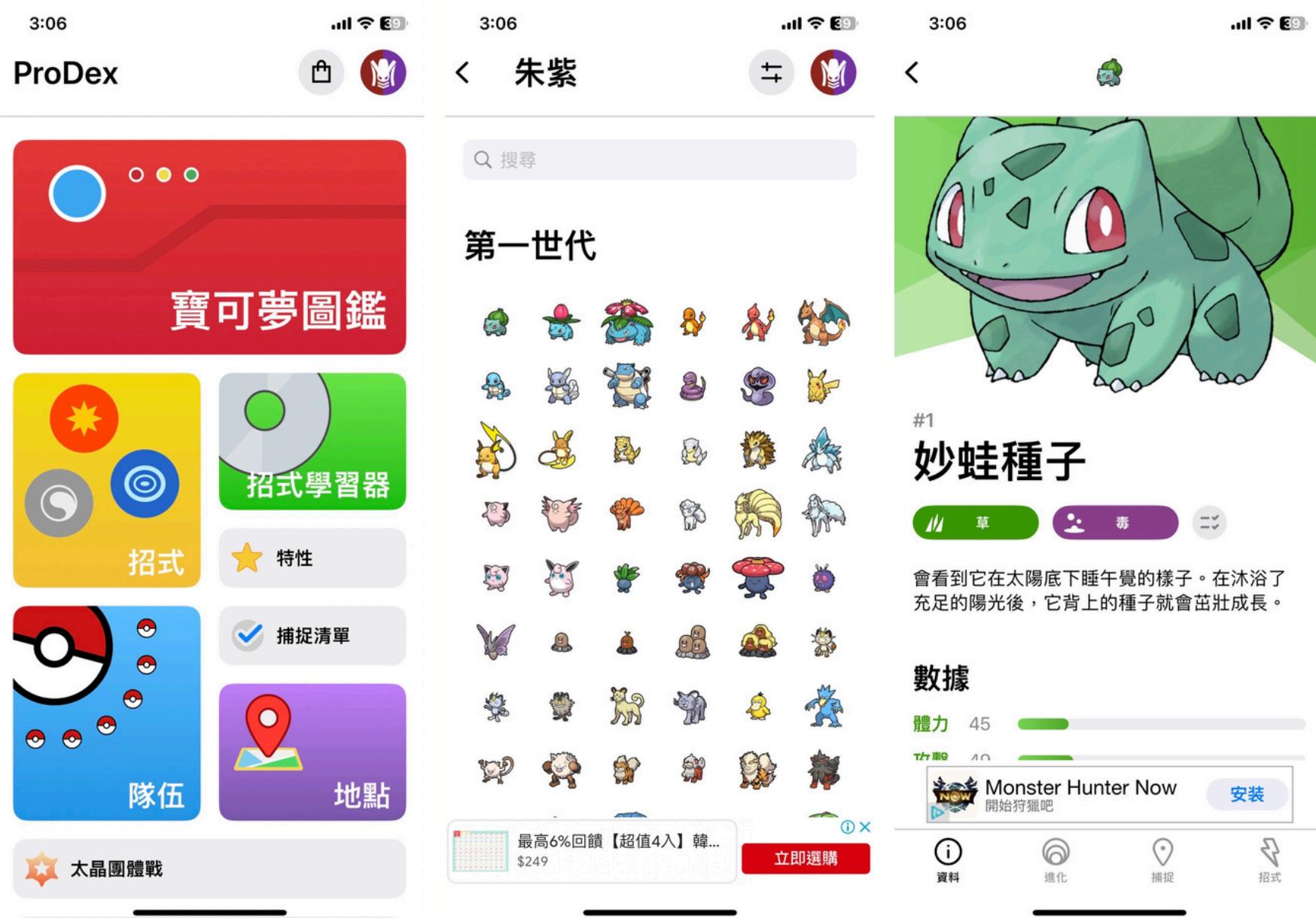
Displays the user's favorite Pokémon, highest challenge score, and app settings.



Inspiration/ Related

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

ProDex

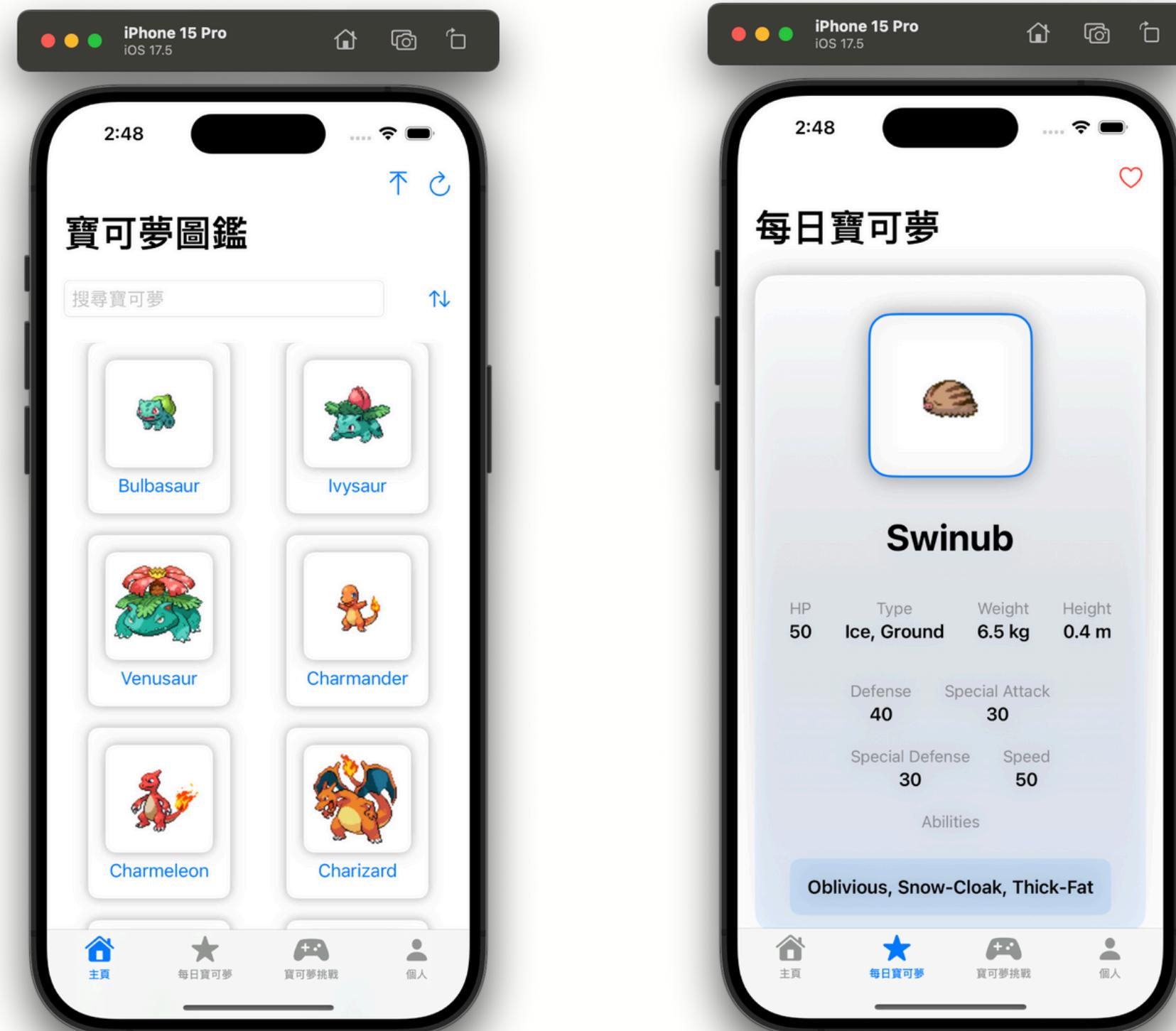


POEMON圖鑑



Introduce Views

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



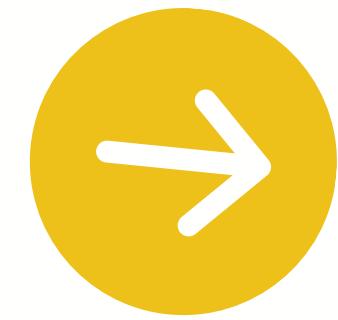
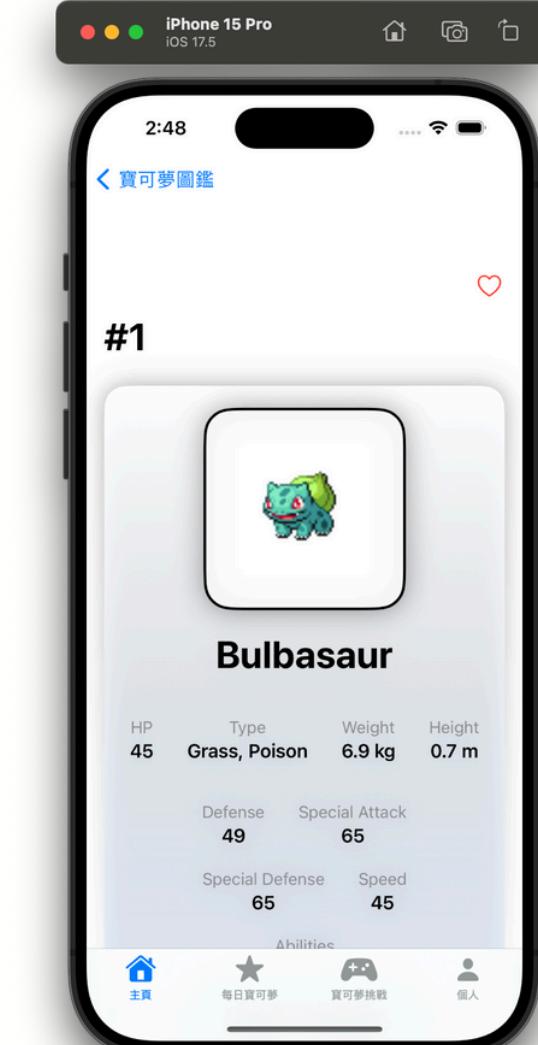
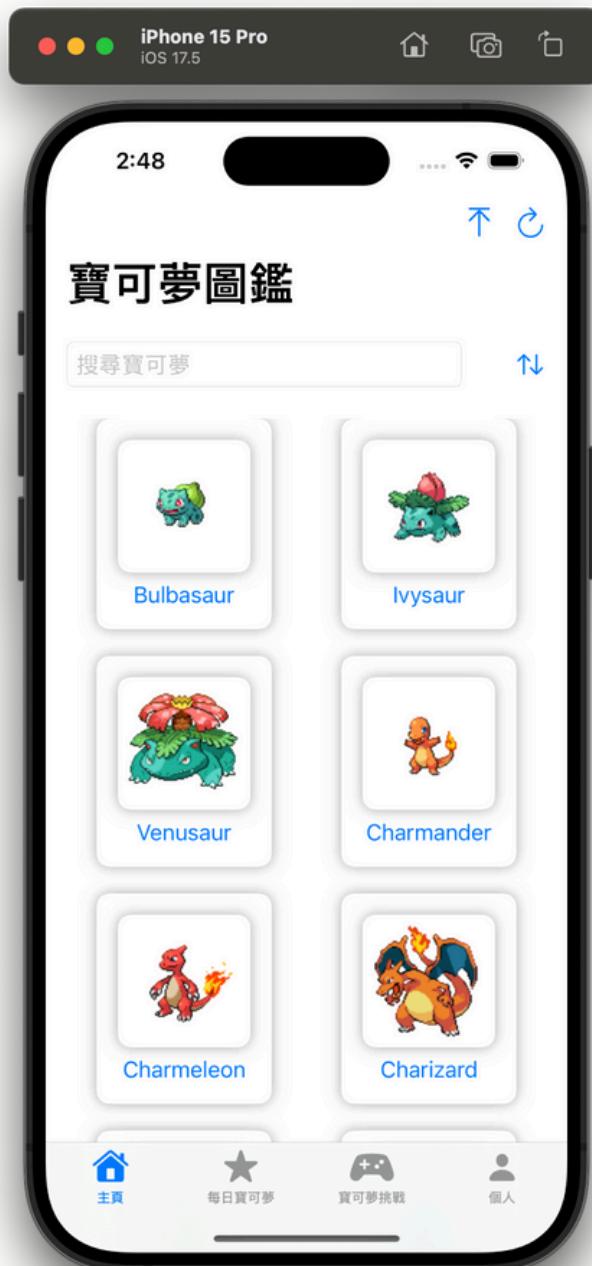
Introduce Views

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



Flowchart

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



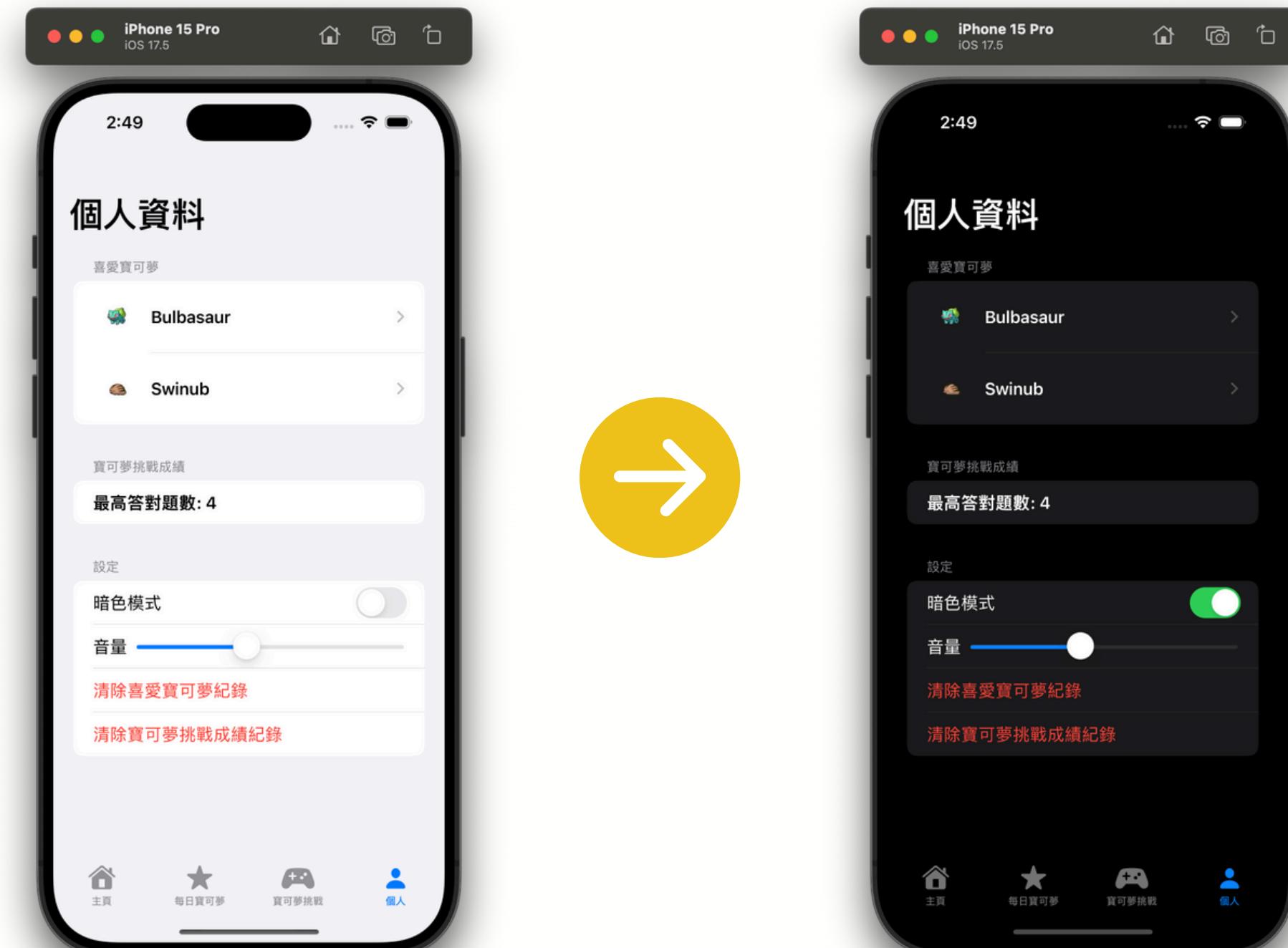
Flowchart

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



Flowchart

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



What part of iOS used?

1

2

3

4

5

6

7

8

9

- **SwiftUI**: For building the UI components and layouts.
- **Combine**: For handling asynchronous data fetching and state management.
- **URLSession**: For making network requests to the PokeAPI.
- **CoreData**: For storing favorite Pokémons and user settings.
- **NavigationView & NavigationLink**: For managing navigation between views.
- **AsyncImage**: For loading images from URLs asynchronously.
- **@AppStorage**: For persisting user settings like dark mode and volume.
- **ProgressView**: For indicating data loading.
- **refreshable(action:)**: For implementing pull-to-refresh functionality.



Discussion

1

2

3

4

5

6

7

8

9

Difficulties Encountered:

- Data Fetching: Managing asynchronous data fetching from the PokeAPI and ensuring data integrity.
- State Management: Keeping the app state consistent across different views, especially with search and sorting functionalities.
- UI Design: Creating a visually appealing and user-friendly interface.

Solutions:

- Used Combine framework for handling asynchronous data flows and state updates.
- Leveraged SwiftUI's data-binding features to keep the UI in sync with the app state.
- Applied consistent design principles and user feedback to refine the UI.



Discussion

1

2

3

4

5

6

7

8

9

Limitations:

- The app relies heavily on the PokeAPI, so any downtime or changes in the API could affect the app's functionality.
- Limited to the data provided by the PokeAPI, which may not include all desired features or details.

Future Work:

- Add more interactive features, such as Pokémon battles or trading.
- Implement a more sophisticated recommendation system for the Daily Pokémon.
- Expand the quiz game with more question types and difficulty levels.
- Integrate social features, allowing users to share their favorite Pokémon and challenge scores.

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



FINAL PROJECT DEMO



2024年
今年も
よろしくお願いします!

