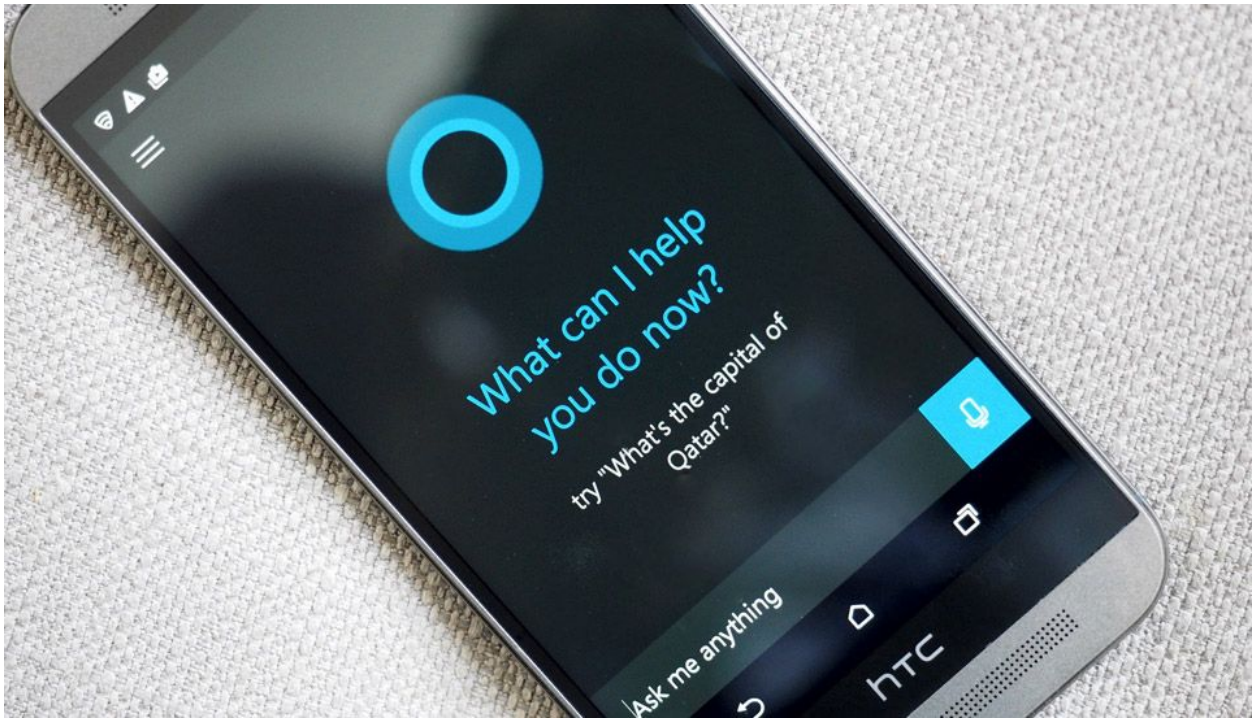

University Assistant using Flutter

Calvin Chow
PCSS Pty Ltd
17 Hardware Lane Melbourne 3000
calvin.chow@outcomelife.com.au
Mobile: +61403318269

Overview

Create a University planner app using Flutter

Goals

Create a voice activated app that provides users with real-time information on their Public Transport Commute:

- Use Google Assistant for speech synthesis and Dialogflow for natural language processing (NLP) to understand what the user is asking
- Make API calls to get relevant response
- Pass response back to Dialogflow which will turn the response data back into natural English that humans can understand
- Finally, output the relevant information through the Google Assistant.

Specifications

Flutter is an open-source mobile application development framework created by Google. It is used to develop native applications for both Android and iOS from a single codebase.

Students will build a university planner app using Flutter that will work on both their Android and iPhones.

The app will be:

- A university planner that will store a student's class schedule, assignment deadlines, and exam timetable
- It will store all the data in a local database
- The app will send push notifications for classes and deadlines and other reminders

The UI of the app is up to the student and they are encouraged to be creative. For example, the main screen of the app may have an "Agenda" screen showing upcoming tasks and

events and a calendar on the top. Items may be colour coded based on the unit it is related to. And a progress screen may show what the student has complete so far.

Eventually, students will be able to update the app to include Firebase to turn it into a fully-fledged app that has login and sync across devices. Firebase is a mobile and web application development platform also developed by Google. Students will be using Firebase's Authentication, Cloud Storage, and Cloud Firestore functions. Firebase has excellent integration with Flutter and has a free plan called Firebase Spark Plan. The Spark plan adequate for the app.

Rationale

Flutter and Dart is a relatively new method of building native apps.

Generally, students who might have taken application development units at university learn traditional methods such as Java and Objective C or Swift to build Android and iOS apps respectively.

Building a Flutter app means learning an entirely new language from scratch.

The idea of the project is to test the adaptability of the student.

New languages and frameworks will come and go every so often. However, the fundamentals of programming remain the same. Students must be able to use prior knowledge of other languages to figure out how to do familiar things in a new environment with a new language.

A project like this will introduce students to the Flutter framework and its various libraries, widgets and other unique features.

Milestones

1. Setting up the environment

a. Set up VS Code or preferred IDE
b. Set up Flutter and Dart
c. Set up Xcode or Android Studio
d. Set up iOS or Android Emulator
e. Set up phone in developer mode

2. Troubleshoot any installation issues students might encounter
    a. Start with basic “Hello World” app
    b. Use the basic app to introduce concepts:
        i. Basic programming concepts to ensure every student is up to speed
        ii. Ensure students understand functions and classes
        iii. Ensure students understand the basics of good UI and UX
    c. Basics of the IDE

3. Use the basic app to introduce Flutter concepts:
    a. What are widgets
    b. Guide students how and where to look up resources:
        i. Flutter Documentation: https://flutter.dev/docs
        ii. Flutter & Dart packages: https://pub.dartlang.org/flutter
        iii. Material Design: https://material.io/develop/flutter/
        iv. Tutorials on YouTube, example apps on GitHub etc.

4. Build app, troubleshoot issues

5. Introduce students to version control and start using git and GitHub

6. Build app, troubleshoot issues

7. Mid-point progress report

8. Introduce students to Firebase depending on their progress so far

9. Prepare app for deployment and installation on real phones

10. Build .apk or put app on TestFlight

11. Presentation and demonstration of app