



By Matthew, Jacob, and Lanndon

Feasibility Presentation Priority To-Do



Language and Framework

Language: Dart

- General purpose
- Developed by Google
- Uses: Web dev, Servers, Desktop, Mobile
- Object Oriented, transcompiles into JavaScript

Framework: Flutter

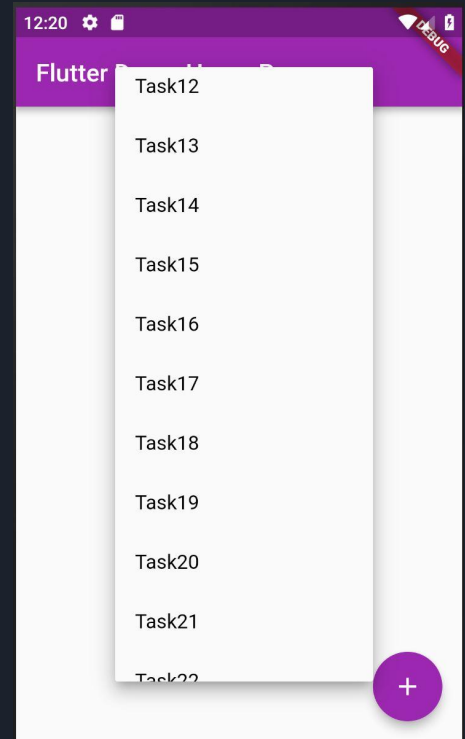
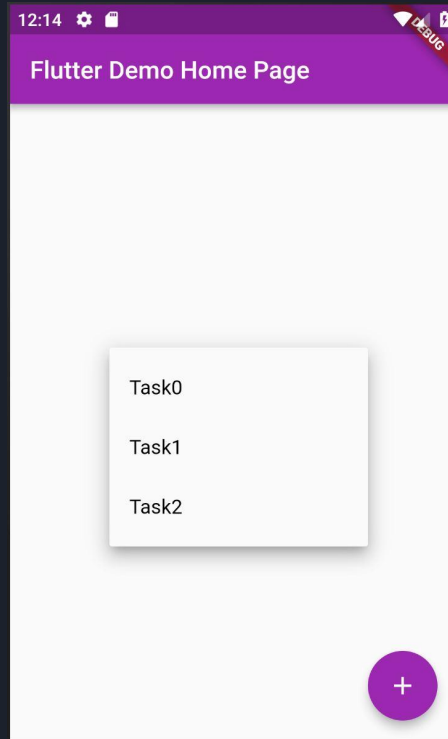
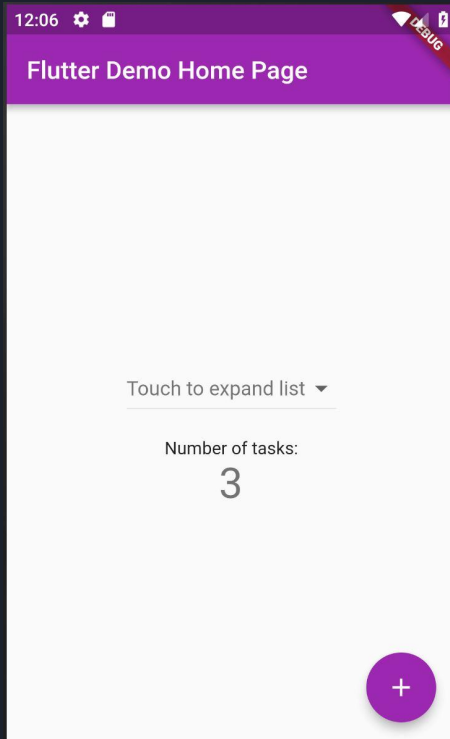
- Open source mobile SDK
- Google contributed to it
- Has small amount of C/C++ in it
- Most feature come from dart giving the developer more control



Repository Organization

- Master will house the working code base
- Each member will make new branch when working on component implementations
- Pull request will be reviewed by all team member before merging to minimize breakage

Screenshots





Sprint 1

We are planning to get the basic functionality of a todo app

- Adding tasks to a list
- Either checking task off or removing once complete
- Recovering task that were check off by accident
- Sorting task based on a priority number
- Have a two page app set up: current tasks and archived tasks



Our choice of Flutter over React Native

- Both Flutter and React Native use languages that are fairly easy to understand/learn. However Dart has ready-to-use widgets which helps ease the learning curve.
- Flutter has more official documentation (Google) compared to React Native.
- Flutter uses its own UI components, instead of third-party libraries, which is ideal with our apps simple GUI.
- The Flutter Console keeps installation/configuration simple, and allows for quick live debugging of our app on either a real device or an emulated one.
- Flutter and Dart have Visual Studio Code extensions to speed up the creation and management of projects
- React Native using native component where Flutter uses proprietary widget sets
- Flutter has better code reuse than React Native when making an app for both ios and android