CSM3114 K1 Project

What To Do

Yet Another To Do List App

Prepared By

GARY LIM KHAI ZHE S62079

Table of Contents

Introduction

This app is inspired by the name itself; not knowing **what to do**, it ended up becoming a to-do list app, but this is probably not enough for the introduction... So, here we go. Procrastination is something that everyone faces at some point in their life, including university students, who delay their tasks and assignments, even though they are fully aware of the consequences of it. Obviously, one of the solutions is to "get good" at time management, and universities could encourage students to have better time management by providing the tools needed to do so, such as a to-do list app.

A to-do list, which is literally just a list of tasks that someone is going to do, is something that people have been using since a long time ago, dating as far back as the 17th century, when someone named Benjamin Franklin used to-do lists in his daily life to make himself better. According to a paper that was published by Michael Pitts and Jennifer Bennett and can be found in the Reference section, writing to-do lists can be used to tackle procrastination. As such, by providing a to-do list for university students, they will have easier access to tools for better time management. Another benefit of having a homebrew solution instead of, say, a third-party to-do list app is that universities can modify the app to fit the needs of university students, providing better incentivization for students to use the app.

Prototype

The prototype will be relatively simple, a barebone functional to-do list app that acts as the base of future improvements. So, what a to-do list app would do?

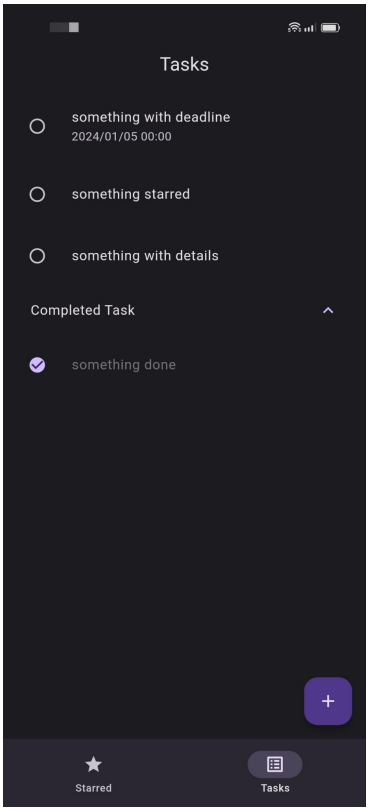
- Add, edit and remove tasks (a task have title, description and deadline)
- Ability to prioritize tasks (add and remove starred tasks)
- Mark tasks as done (vice versa)

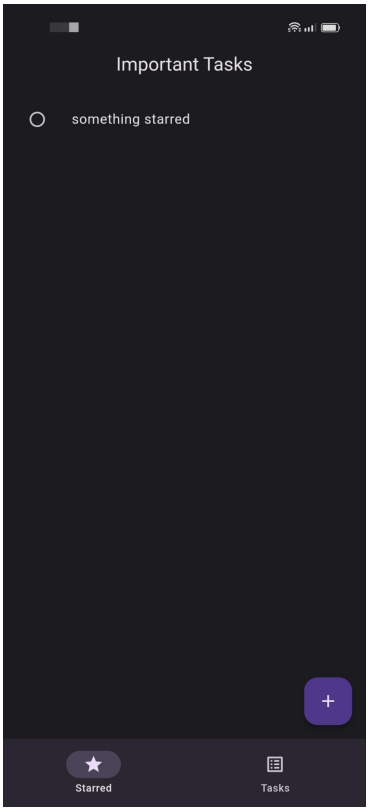
For the implementation, Flutter will be used, and we won't explain why Flutter is the best framework for app development whatsoever, since anything goes, so Flutter it is. Now obviously, a Task object will be used to keep the properties (listed above), and speaking of data, you probably (totally) would want the app to keep the data even after the app is closed, which means, SharedPreferences will be used. For the structure of the app, it is roughly as the following:

MaterialApp

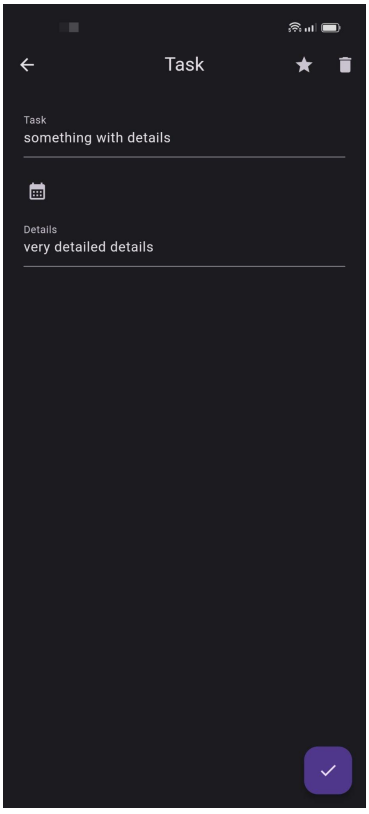
- HomePage
    - StarredTaskList (ListView, starred task)
    - TaskList (2x ListView, todo and done)
    - FloatingButton (a showModalBottomSheet to add task)
    - NavigationBar (travel between tasks and starred tasks)
- EditTaskPage (accessed from tasks and starred tasks)

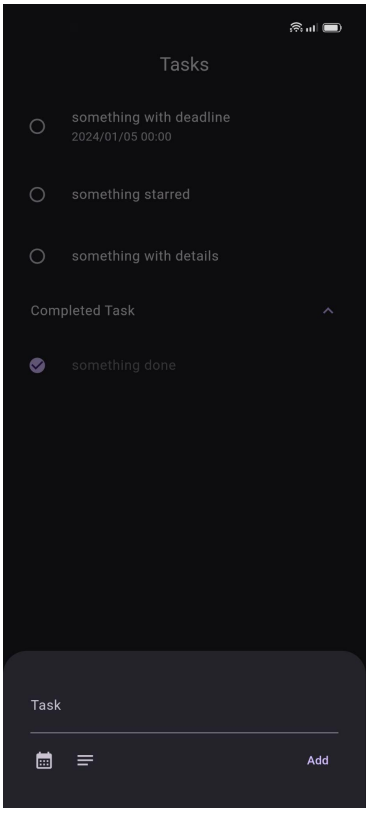
User Interface



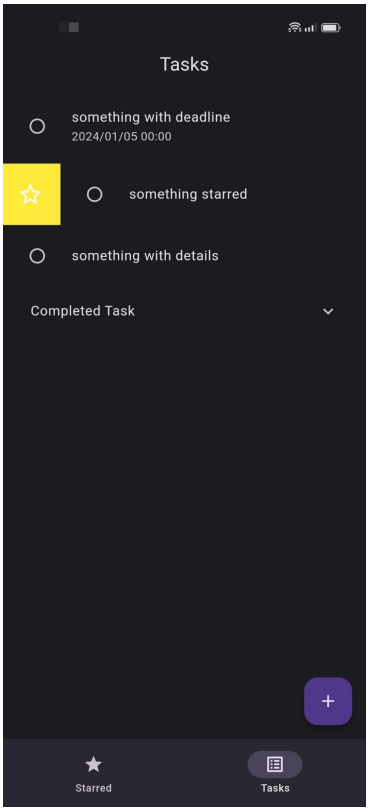


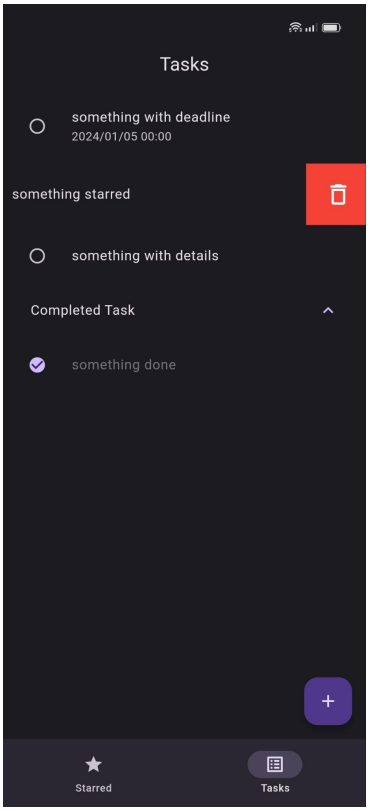
- home page, todo, done & starred



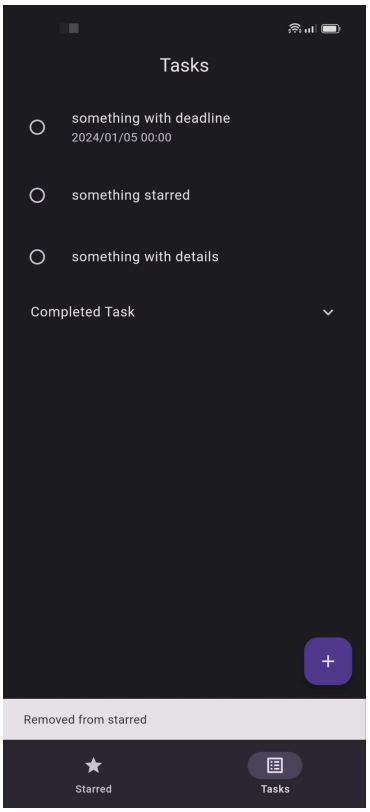


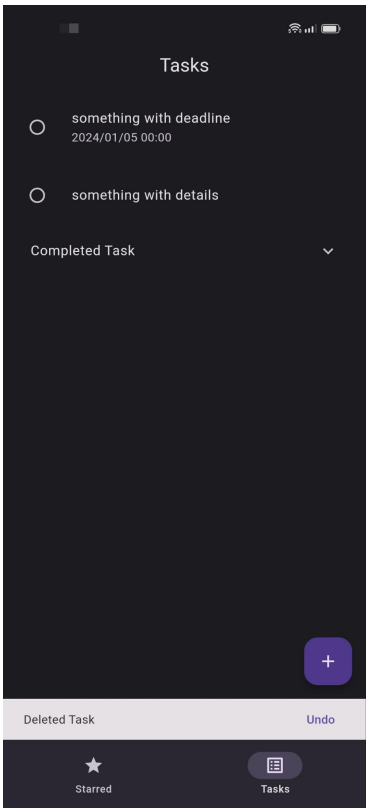
- add task, edit task (+)

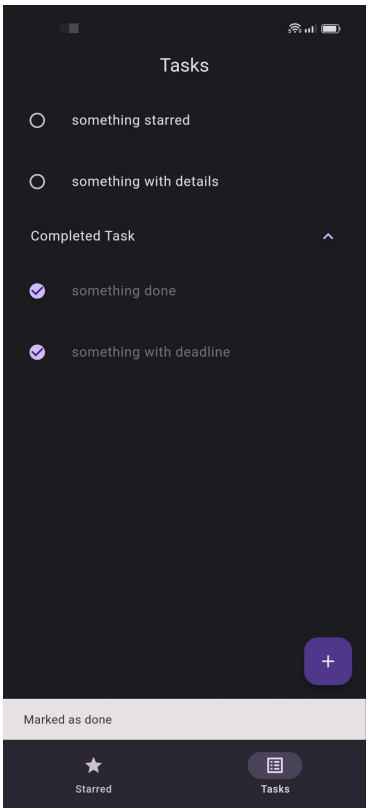




- gesture to delete and star task







- snackbars to notify and undo actions (where applicable)

Pricing

Generally speaking, due to the nature of the app being part of free offerings of students for their self improvements, and partially due to the fact that this is a university incentivized product, the app itself would likely be free. Therefore, there will be no pricing, since there will be no server to maintain either. That said, even if the app decides to add features that are more university oriented (such as an anonymous social tab that allows students to publish their finished task and interact with others posts), it will most likely be subsidized by the university anyway.

However, this would ironically also mean that the app itself has little to no commercial value. As such, we can include some monetization method that either pissed user off or convince them that it is a good idea to pay for the app, such as the inclusion of ads in "free" versions of the app and making them pay for no ad experience, locking some more useful features such as searching task (for whatever reason why anyone would want to do that) or cross platform data syncing behind a paywall. With these possibly despicable additions, we might be able to milk some cash out of a minority that enjoyed the app, while the rest would probably go for the free alternatives such as Google Tasks and Microsoft To Do, which have those features aforementioned but without the ads, since they monetize in a different way.

Lesson Learned

Language wise, I'm exposed to a lot of new Widgets, such as SharedPreferences, Inkwell, CheckboxListTile and some more. Even though some did not end up in the final prototype, I had a lot of fun messing around with those widgets. For the development lifecycle,

I have realized that agile development is a good approach for app development, since it is more iterative and faster way of verifying functionality, but it also comes with the obvious weakness where you will start to have difficulty in traversing the project due to the constantly expanding codebase and it will starts to get real messy, which is a pretty common occurrences in OOP development. While some argue that you could have done something like separating the code into modules, employing different programming disciplines, the truth is that you likely won't know what you need until you start doing it. Also, if we have to plan out everything from structure to code, that's literally a waterfall development.

Conclusion

In conclusion, we have planned a to-do app, and successfully created it, learned a lot of useful and occasionally niche knowledge along the way and made sure that the app is actually functional. I won't end this conclusion with something generic like "I hope this app will be successful and help many people", because realistically, it doesn't, and won't achieve those goals. But yes, it could become something that I personally would use for task tracking.

For myself, the biggest takeaway from this project is that, if you want to make something, make sure it is functional, and you would actually want to use it as an app, because it would be pointless then if that is not the case. Lastly, remember to have fun, even sometimes it feels like it sucks (burnout or whatever), also, don't forget that you have other projects (that you haven't started yet) and spent too much time on one project.

Reference

1. Procrastination on Wikipedia, https://en.wikipedia.org/wiki/Procrastination
2. The Surprising History of the To-Do List and How to Design One That Actually Works by Belle Beth Cooper, https://buffer.com/resources/the-origin-of-the-to-do-list-and-how-to-design-one-that-works/
3. Just Do It: Tips for Avoiding Procrastination by Michael Pitts and Jennifer Bennett, https://inside.sou.edu/assets/socsci/Advising__Student_Success/Motivation__Procrastination/kaavya-just-do-it-tips-for-avoiding-procrastination.pdf
4. https://api.flutter.dev/flutter/material/NavigationBar/backgroundColor.html
5. https://api.flutter.dev/flutter/material/NavigationBar-class.html
6. https://docs.flutter.dev/data-and-backend/serialization/json
7. https://stackoverflow.com/questions/65141485
8. https://stackoverflow.com/questions/45948168
9. https://stackoverflow.com/questions/70830642
10. https://stackoverflow.com/questions/58522998
11. https://stackoverflow.com/questions/54515186
12. https://stackoverflow.com/questions/53869078
13. https://stackoverflow.com/questions/51012360
14. https://stackoverflow.com/questions/55211640
15. https://stackoverflow.com/questions/73669989
16. https://stackoverflow.com/questions/60314623
17. https://stackoverflow.com/questions/62927579
18. https://stackoverflow.com/questions/55777213