

Checkpoint 2: App States

This is a short one. We want you to think about the various states that your app will be in. What needs to happen when your app loads? When it will become inactive? Does it need to do work in the background? Also, consider how these states change based on other factors. What if there's no internet connection? What about when the user isn't logged in?

1. Describe the various states for your app. For each state, think about what needs to happen programmatically, what the interface looks like if any, etc. The more detail the better.

When the app loads, we will prepare the landing page (timeline for now, floating texts once it's added). The app will become inactive when the user moves the app to background. The only work that needs to be done in the background are 1) data storage and iCloud syncing, 2) potentially auto indexing for faster search, 3) potentially entry-based auto recommendation, which is an R&D feature. The app should not be significantly influenced by other factors: the internet connection is not needed for user supporting user interaction (it would be needed mostly for iCloud syncing and recommendation), which takes place in background/asynchronous queue only if there's internet connection, and we do not require user login.

State Transitions

Unattached State

- ☐ Prepare to get data back from iCloud

Foreground Inactive (Launch Screen UI)

- ☐ Prepare the landing page (timeline for now, floating text view once it's added)
- ☐ Data can be retrieved from iCloud to prepare for the **Active** state

Foreground Active

- ☐ Update data model with the synced iCloud data
- ☐ Switch between different views (timeline, detailed view, etc) depending on user interactions (respond to user actions)
- ☐ Update or add data entries based on the user's action of editing or creating new title entries. Attempt iCloud syncing
- ☐ Continue dispatch queues from API calls

- ❑ If the user exited the app while in the editing page, and the app returns to the foreground before being detached, resume the previous unfinished session.
- ❑ After we prepared all the data, we can then show the UI to the user, and our app becomes active.
- ❑ (R&D) Recommendations can be updated when new user input is recorded
- ❑ (R&D) Auto (re)indexing for added/edited entries

Foreground Inactive (arriving from Foreground Active):

- ❑ Save data and attempt iCloud syncing, if not already done

Background:

- ❑ Pause any dispatch queues and release memory
- ❑ Prepare for App snapshot (hide entry details according to user settings)

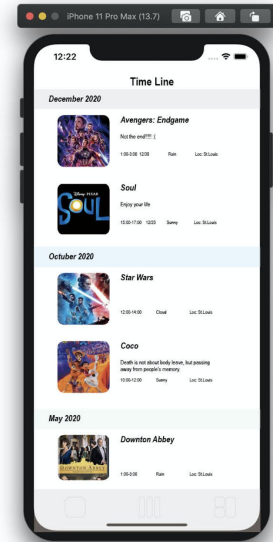
next page...

View Content

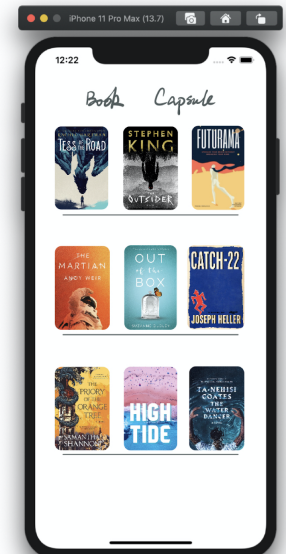
Launch Screen (Main Page):

If we successfully implement floating text, then the launch screen will be showing up the names of all the movies/books/games the user has finished. The names will float from the right to the left. The more stuff the user finished, the denser the screen will be (which gives the user a sense of accomplishment).

- ❑ Load the movies/books in timeline, for the basic version, it will be a table view, each day starts with a time indicator, and then comes the movie/book finished that day.
- ❑ For each table view cell, there will be a picture of the user's choice, a title, a rating, and maybe also the weather of the day.
- ❑ Add Button: there will be an “add” button on the page, probably on the top right or maybe in the tab bar, which will allow users to create a new note.



- ❑ Tab Bar:
 - ❑ First Item: show timeline
 - ❑ Second Item: show the bookshelf view if implemented
 - ❑ Third Item: tap to navigate to create the seasonal/yearly summary for the user
- ❑ Click on a specific cell to get into the detailed page
- ❑ Local search on existing entries



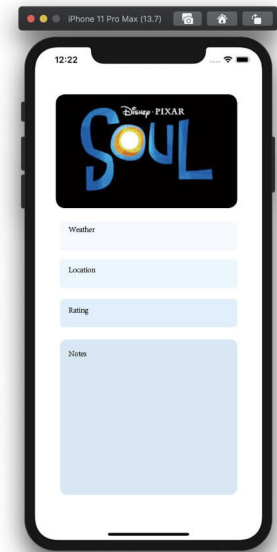
Detailed Entry (accessed from main page):

1. Load the detailed information of a specific movie/book/game, including the picture, weather, rating, and user's notes.

Add Entry

Connect the content which the user is typing in with the data from some API such as Hulu to do smart search.

- ☐ A button for user to upload a custom picture
- ☐ A field that allows users to add/edit their comments
- ☐ Rating Bar



Summary Page:

- ☐ User can create a poster-like summary
- ☐ A button switching between monthly/yearly view
- ☐ Able to navigate back to Main Page