# Roblox

## Readr: A Micro-novel Reading App

*The Department of Literature is building an app to encourage people to read more for fun. This app will randomly push micro-novels according to users' chosen mood, types of interests and estimated reading time.*

This app is called Readr, because it's a reader but shortened!

## Research & Considerations

### Why do we need a micro-novel reading app?

To fully design engaging features and expand on this idea - I took to informal user research and secondary research (sources <u>here</u> and <u>here</u>)

**Users want to read shorter novels because:**

1.  Lack of time for fully-fledged novels

    a.  Need and desire to finish stories - a longer form novel would make users feel guilty for not being able to finish

2.  Lack of attention span - rise in the consumption of short-form media

    a.  There are other micro-reading apps - such as **news apps** and **webtoons (micro-comic apps)** that are on the rise because of their bite-size format

3.  When would you use this app

    a.  While doing morning errands, when riding a bus or a train

**Users are concerned to read because:**

1.  Fear of poor content

    a.  "What if the stories are boring and a waste of time?

    b.  "I don't want to read poor quality stories because I don't have a lot of time"

  c. Would be cool if it were selected by someone noteworthy

2. Lack of consistency and motivation

  a. "I'd only read to sound intelligent in conversations"

  b. "I don't think I could read a story every day"

3. What if I don't have time

  a. get to contact as fast as possible: Has specific favorite genres

# Product Thinking Exercise

To go beyond the scope of a novel app, I brainstormed and sketched several concepts to take it beyond. Assuming that there are engineering constraints - I narrowed down to top features I'd implement. Below, the ideas are listed from most feasible (1) to least feasible.

## How do we...

### Allow users to trust the content?

1. Already have high quality content pre-curated ✅

2. Allow author bios ✅

3. Have carefully curated content lists ❌

### Allow users to find content as fast as possible?

1. Allow filtering ✅

2. Short cuts for common filtering ? ❌

  a. Would require more brainstorming to figure out what common shortcuts would be

### Motivate users to use the app?

1. Progress bar to reach "Reading goals" ✅

  a. ~~Daily streak goals or~~ monthly goals - because consistency over a week is harder than reaching one goal a month

2. Allow quote and content-based sharing systems ❌

  a. Works for instances like medium where pieces are more of "think pieces" - while these articles will range from factual to purely entertainment, making less sense

for quotes in the MVP

3. Streak system ❌

4. Point system ❌

   a. Any point/streak system would require extra engineering that would be difficult to do and doesn't ensure user retention for all use cases

# Information Architecture

**Functions**

1. Users want to read relevant literature

   a. Filter function to query the right type of novels

   b. Way to edit the filter

2. During preview time - users want to know which to read

   a. Summary

   b. Graphic

   c. Tags indicating type/reading time/mood

   d. Author bio

3. During reading time - users want to share it and read more

   a. Recommendation system, save and share system

**Pages**

1. Homepage

   a. Filters

   b. Recommended readings

   c. Different scanning settings (limited vs full preview)

   d. Progress bar

2. Readings Detailed Page

   a. Text

b. Author bio

c. Save/share

d. Recommended readings to look at next

    i. Reading preview: title, picture, tagline, time, mood + type recommendation

    ii. Author bio

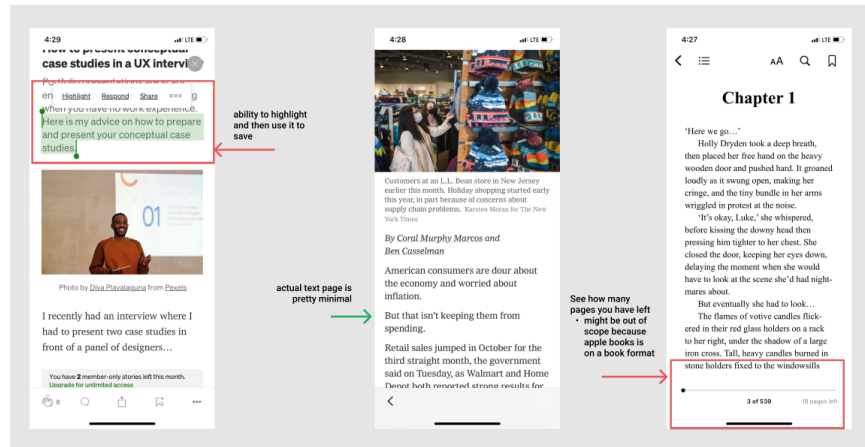3. Saved

a. Completed readings

b. To read later

# Competitive Analysis

After doing a competitive audit of other apps below - I took inspiration on aspects that I found worked well and others that didn't, as well as the contexts they existed in. Red indicates a design pattern to stray away from, while green indicates one to keep.
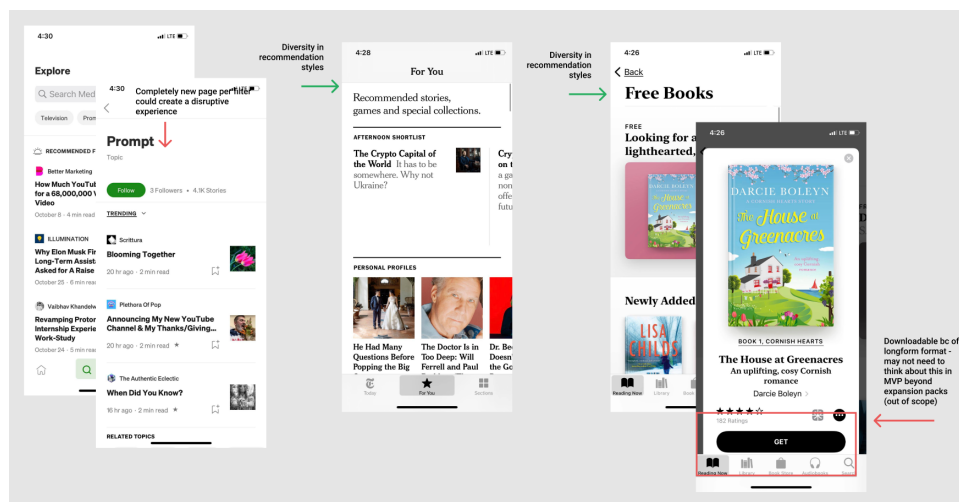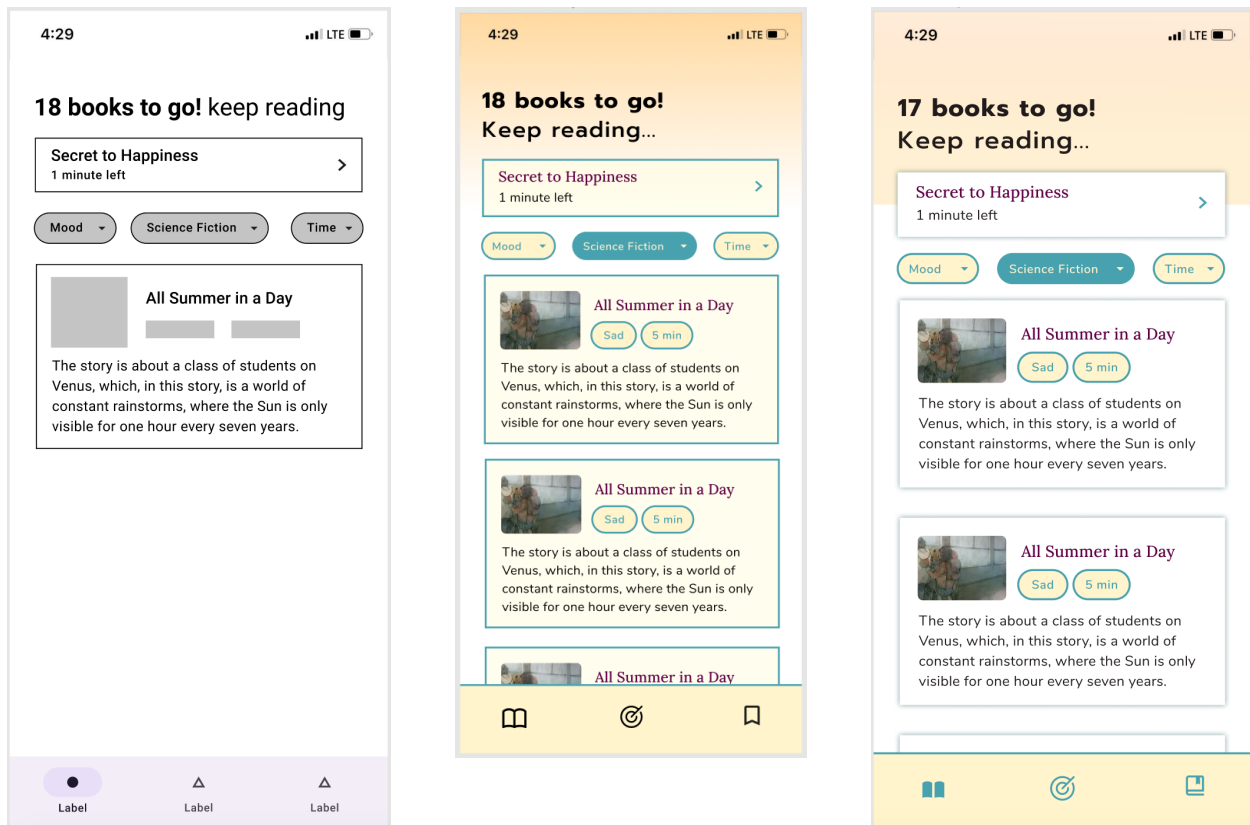
## Homepage
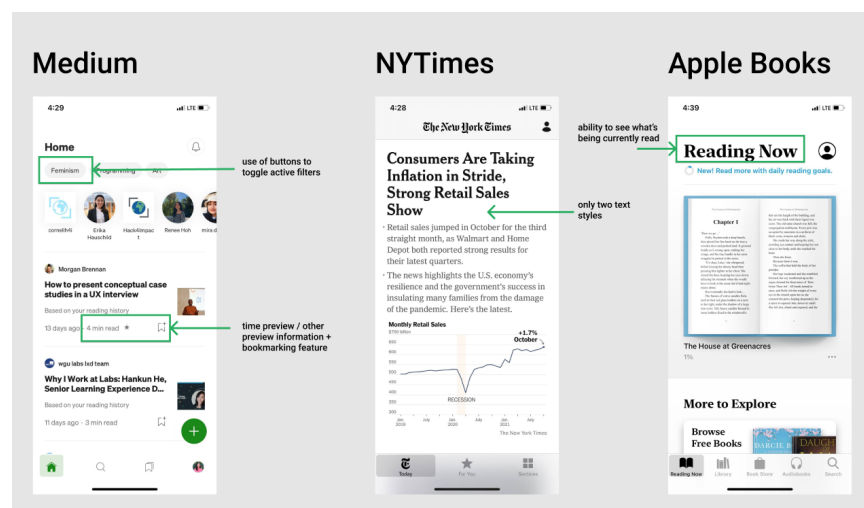


## Reading

## Browsing



# Iterations

## Homepage
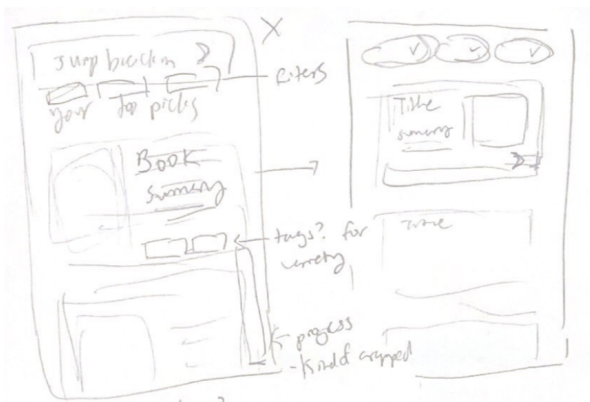
For the homepage - I was initially unsure whether or not to have a keep reading to allow the reader to jump back in and also the progress bar. I eventually decided that the progress bar made the homescreen too cluttered and dedicated a separate section to it.
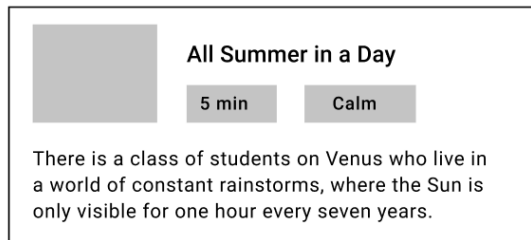


For the iterations, as I went up in fidelity, I realized the blue borders were overwhelming and claustrophobic. I decided on keeping to drop shadows to allow distinction between cards but also allow it to be minimalistic.
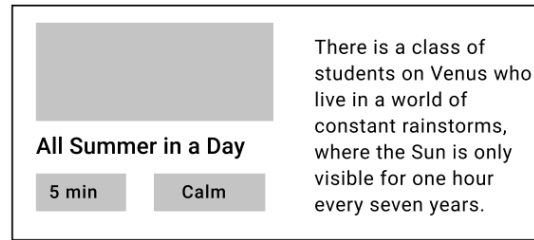
## Card

For the actual card I went through multiple iterations, I decided on a variation of version A because this allowed for the most freedom in text as well as include enough space for the tags (especially since there were use cases of three tags). Most pictures wouldn't fit the aspect ratio in C and D so I decided against it.
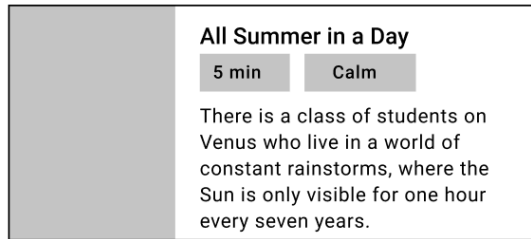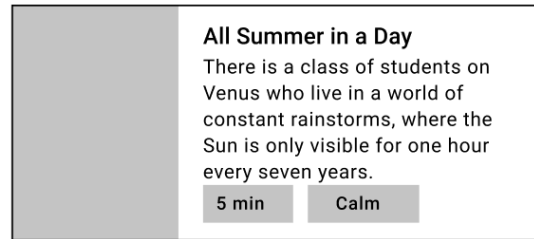
**Version A**



All Summer in a Day
5 min    Calm

There is a class of students on Venus who live in a world of constant rainstorms, where the Sun is only visible for one hour every seven years.

**Version B**



All Summer in a Day
5 min    Calm

There is a class of students on Venus who live in a world of constant rainstorms, where the Sun is only visible for one hour every seven years.

**Version C**



All Summer in a Day
5 min    Calm

There is a class of students on Venus who live in a world of constant rainstorms, where the Sun is only visible for one hour every seven years.

**Version D**



All Summer in a Day
There is a class of students on Venus who live in a world of constant rainstorms, where the Sun is only visible for one hour every seven years.
5 min    Calm

## Read Details

For read details I decided to have the user indicate when they finished reading because it was an easy engineering way to figure out when they were done with the story. I included an "also check out" section because we want to keep the user on the app for as long as possible. It's possible their commute or ride takes longer than the allotted story time, meaning that it would be a good idea to suggest more stories.
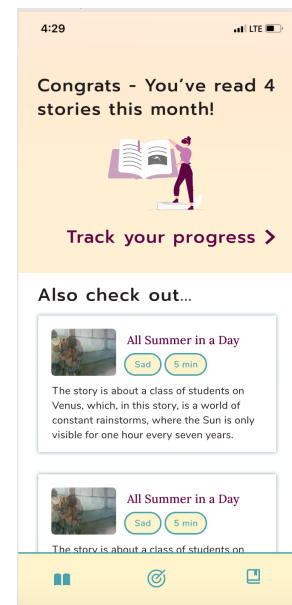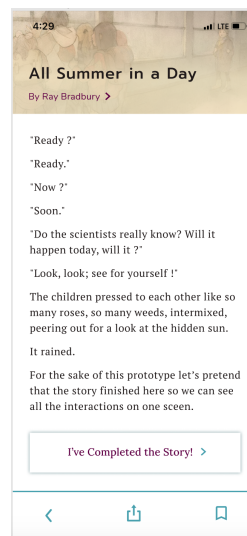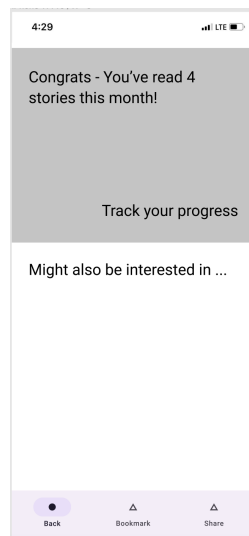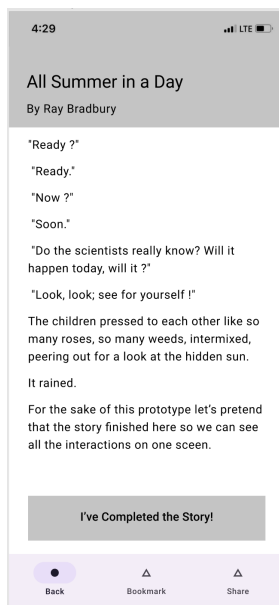
## Saved
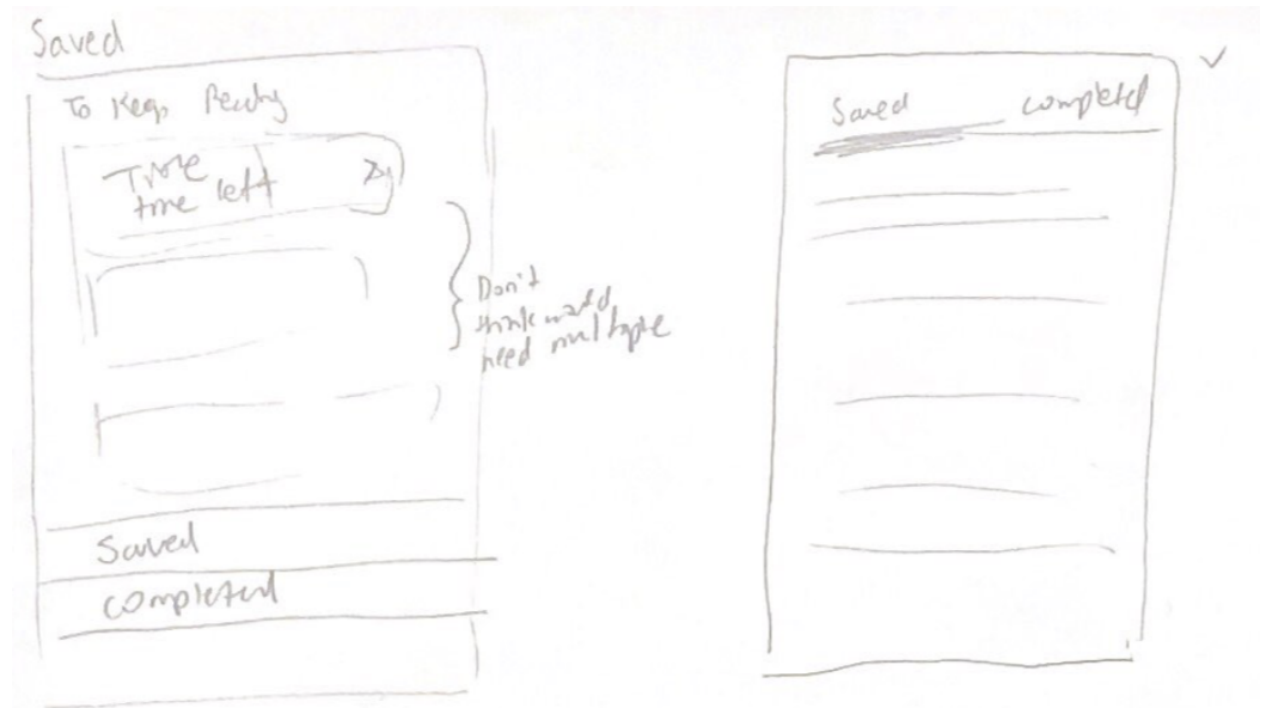


For saved, originally I was going to have saved, completed, as well as current reads. I realized that users will not be reading multiple short stories at the same time, so I scrapped current reads. I eventually decided on including a goals page, which would make more sense to have the history there because you'd want to see what books you read to get to your goal.
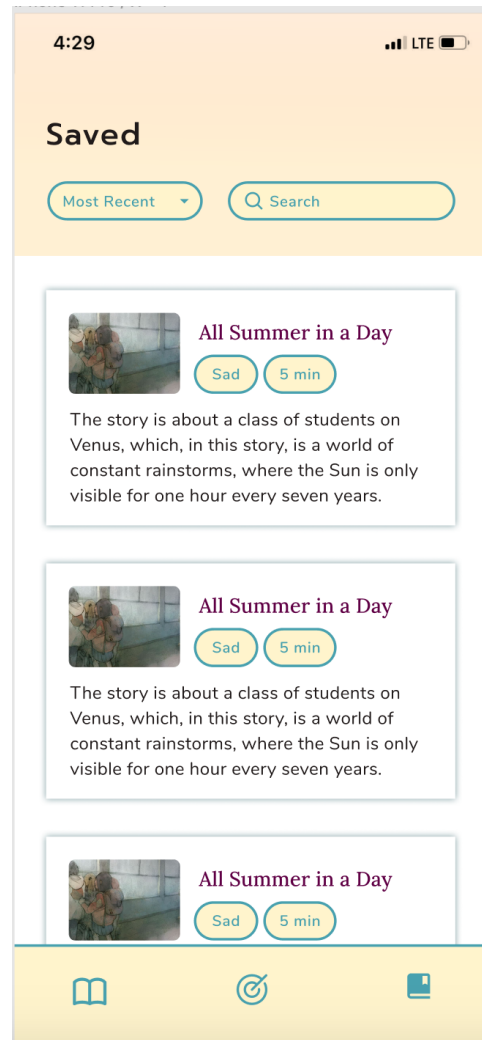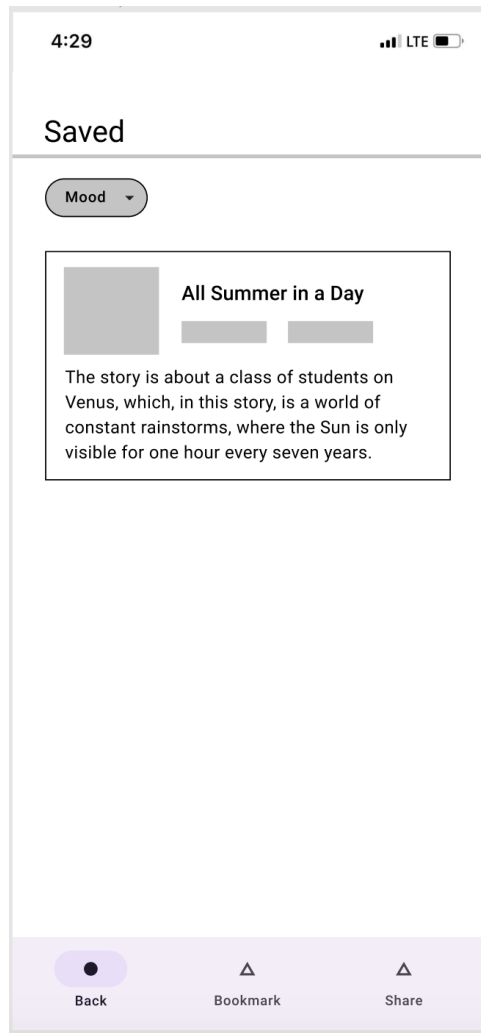
For saved I decided to allow sorting as well as searching for a specific story they remembered and were interested in. This would allow the greatest amount of organization without having to make a new feature like stories.

## Goals
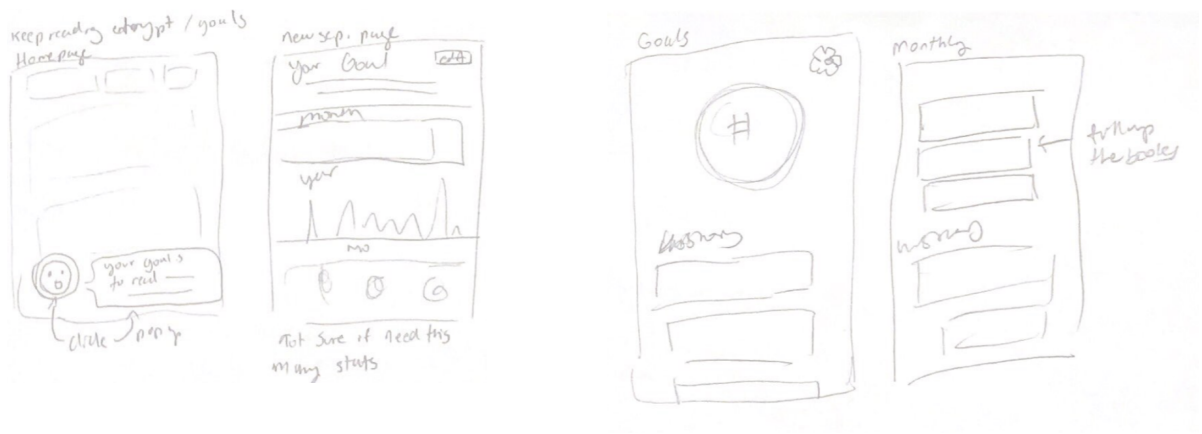
For the goals section, I thought of multiple ways to display the statistics, even considering having a chatbot icon in the front of the page to encourage the user to read. However, I figured that user would remember the goals entrypoint easily and wouldn't require this extra prompting. I settled on a separate goals page with one visualization to show users how close they were to completing their short-term goal.
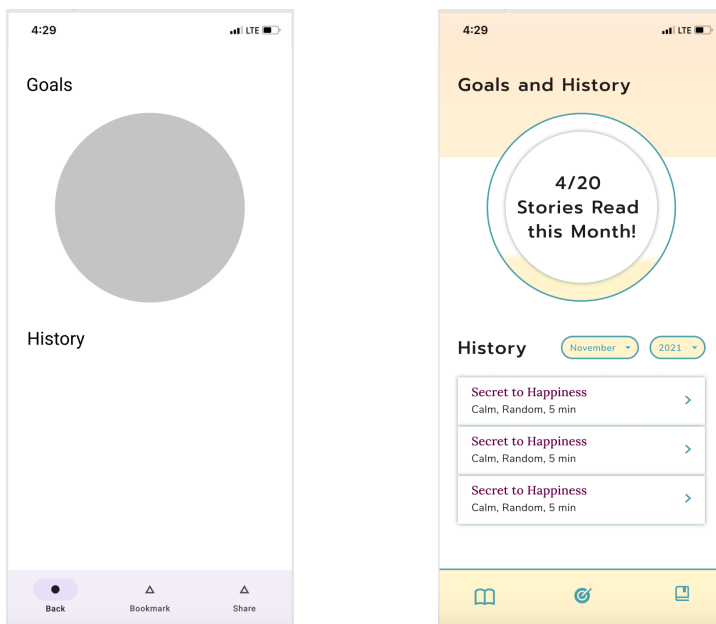


For the UI of the history, I did not have the lengthy preview cards as before. This is because users most likely remember the plot of their most recently read stories so I decided it would be redundant.

## Final Prototype
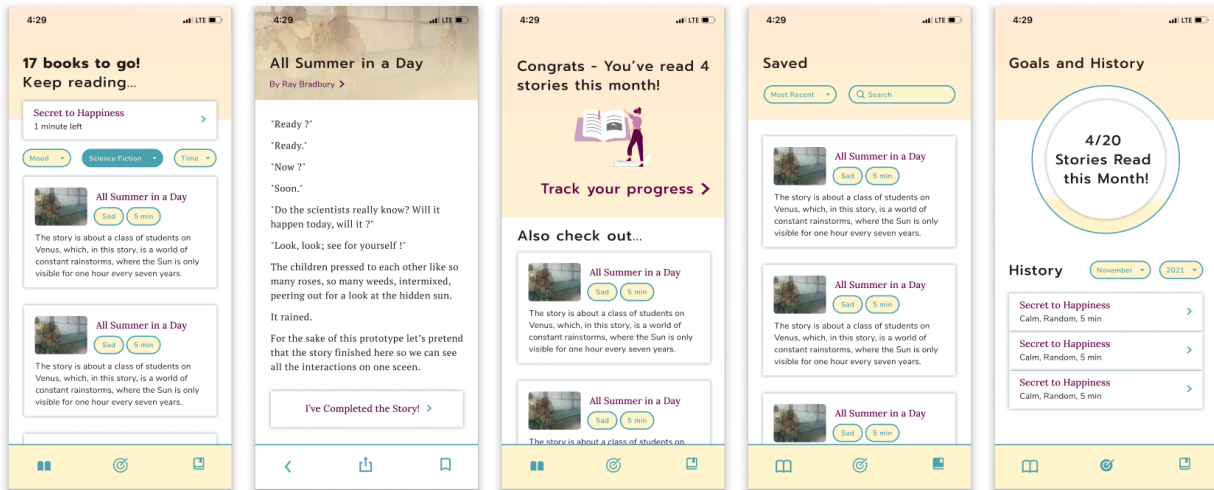
# Reflection

To reflect on this journey, I feel like I spent a lot of time making small design iterations that weren't documented well. If I had more time, I would focus more on making mid-fidelity iterations however for times sake I settled on doing them in low-fidelity instead. Although effective effort-wise, it was difficult to distinguish details and communicate them easily should I present these designs during a critique.

Ultimately, I'm proud of the amount of product thinking that went into feature determination. Also, if I had more time I would play around with the branding elements a bit more because the colors are heavy on the different shades of yellow. All in all, this was a fascinating design exercise that challenged my presumptions on what I intuitively knew as usable. It also challenged me to iterate quickly directly mostly in high-fidelity which required much Figma shortcuts.