James Dishman
February 15, 2025
UX 60503
Assignment: Mobile Design Patterns

Part 1: Explain Mobile Design Patterns

In Part 1, we will look at common design patterns in mobile applications. In Part 2, we will examine examples of these design patterns in current mobile applications.

1) **Navigation**
    i) **Sliding drawer (similar terms: side menu, navigation drawer, hamburger menu, slide-out menu)**

        The sliding drawer is a type of menu. It is commonly referred to as the "Hamburger Menu" because it is widely displayed as three horizontal lines on top of each other, reminiscent of a hamburger. Its primary purpose is to hide secondary navigational functions, leaving as much screen real estate as possible for the application's primary purpose. Sliding drawers are helpful for more complex applications with many secondary options that users might only need to access periodically, and the application must maximize available screen space.

        The sliding drawer can declutter the limited space available for an application. Some problems with the sliding drawer are that it may be less discoverable for users and increases cognitive load for users who may not know what is inside the drawer and when to use it.

    ii) **Upper area tabs (top tabs, top navigation tabs, tab navigation)**

        In most cases, upper area tabs are displayed across the top of the screen Flowing left to right for English reading users. They separate content into categories and divide the application into sections. Historically, upper-area tabs have been more prevalent in Android applications. You can commonly see this type of navigation deployed in news, e-commerce, social media, and travel booking applications, where the application is served by being divided into different sections.

        Upper area tabs are visually clear and divide the information into accessible sections. They are flexible in that the copy and iconography can be tailored to fit the application's design. Ergonomics can be an issue with upper area tabs, as the user will need to stretch the thumb across the application to utilize them.

    iii) **Bottom menu (bottom navigation, bottom bar, bottom tabs, navigation bar, tab bar)**

The bottom menu has historically been the dominant design in iOS applications but is becoming more prevalent in Android applications due to the natural ergonomics of phone cradling and the use of the thumbs to navigate mobile applications. It makes sense to have the available options closest to the thumbs, which places the navigation bar at the bottom of the application. The bottom menu is also easier to interact with when using the phone one-handedly. This position leaves the upper screen real estate available for other information.

Ideally, a bottom bar will have five or fewer sections to allow for adequate screen space for the thumbs to access each section. There can be a method to access "more" sections if necessary. This is often displayed with three dots. Bottom menus are more user-friendly and ergonomic than having tabs at the top of the application, which has led to its prevalence in application design.

iv) Floating Action Buttons (FABs) (Floating button, quick action button)

Floating action buttons are hideable buttons that hover over the content displayed on the screen. They are ideal for applications where the information takes up the entire screen and may need to be manipulated frequently with gestures. FABs are often deployed towards the bottom of the screen, where they obscure less information and are easily accessed with the thumb. FABs are ideal for primary actions that may need to be used repeatedly. FABs are generally seen most often in Android applications, but they are appropriate for any application that displays large amounts of changing data on most of the screen.

2) Content Layouts

i) Filmstrip

A filmstrip layout is practical when developers must display conceptually related information that needs to be distinct. The user generally swipes left to right or utilizes arrows to switch panels and compare information. Typical examples are weather in different cities, sports scores or stats for other teams or sports, pictures that display one at a time, or a reader that displays one page at a time. Ideally, a filmstrip would contain only enough screens that the user could compare what they need to compare without excessive swiping between screens.

ii) Collections and Cards (card layout)

Collections and cards are a way to display multiple pieces of content at the same time, which allows the user to compare the content and select what they want. Collections are generally thumbnails, while cards can contain a thumbnail and more information. Collections would be ideal for displaying videos, while cards would be more appropriate for blog entries, articles, or lists of applications. Both are ideal in a mobile setting as a method of relying less on text, which could be challenging to read in some settings. They are generally grouped in a way that allows multiple entries to be displayed on the screen so that users can pick whatever is appropriate.

iii) Infinite list (infinite scrolling, continuous scrolling, lazy loading list)

Infinite scrolling has become the dominant method of delivering content to users for multiple reasons. Infinite lists keep the user engaged with content, but they can allow for smaller amounts of data to be loaded in increments, keeping the experience fluid by eliminating or reducing load times. This is known as lazy loading. Page loading buttons are available to load additional results pages and are commonly employed by search engines. This approach is a helpful way to provide the appropriate amount of information to the user with limited screen space.

Part 2: Identify Examples

1) Navigation

a) Sliding drawer (similar terms: side menu, navigation drawer, hamburger menu, slide-out menu)

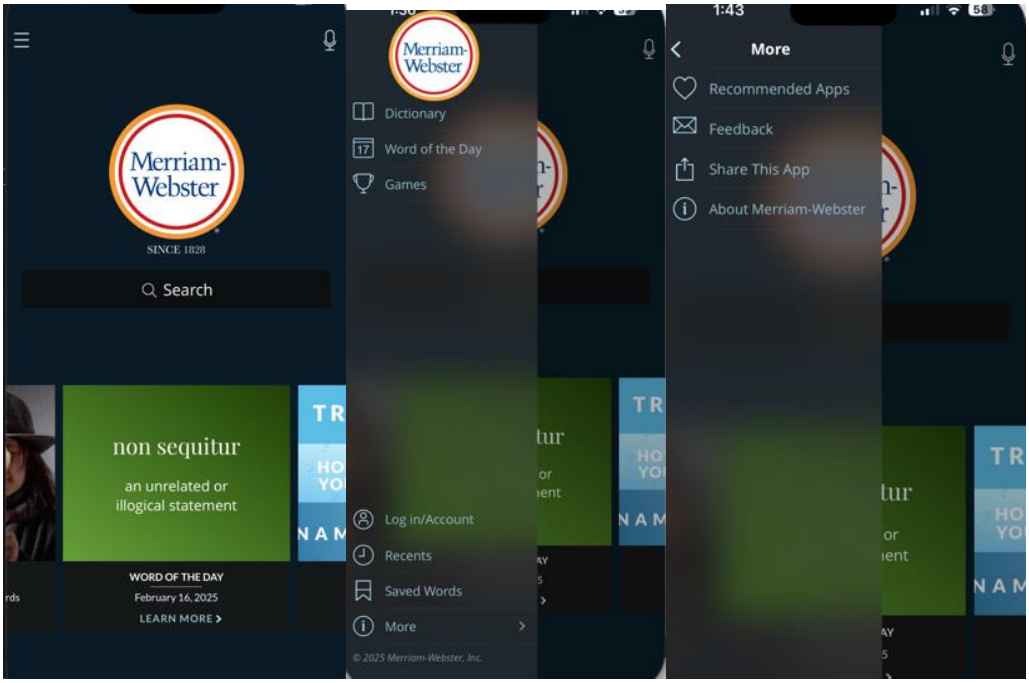

Figure 1 Sliding drawer in Merriam-Webster.

Here is an example of the sliding drawer; when the three bars are pressed, it activates secondary functions that are used less often. The application's purpose is to look up words, but the Word of the Day and Games section is also popular. Log In / Account, Recents, and Saved Words would be used less often, hence the hierarchy. The options in the More section are used even less frequently.

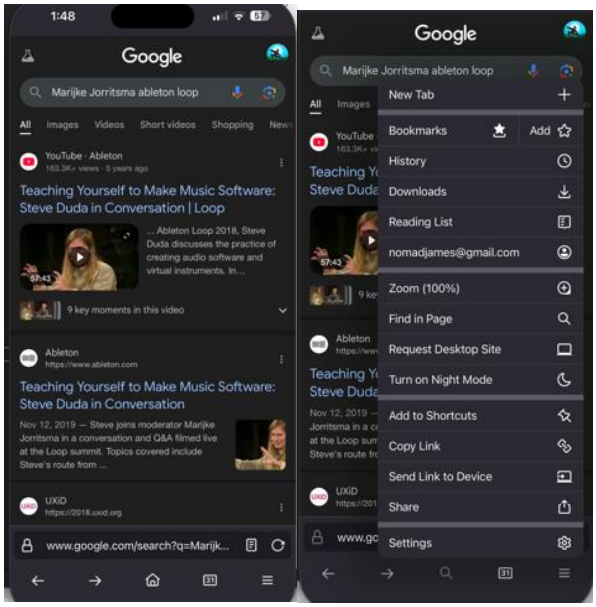

Figure 2 Sliding drawer in Firefox.

Here, you see another example of the sliding drawer. In the case of Firefox, it is located in an abnormal location, but it is still very easy to find and understand its purpose. There are also many more options on the Firefox menu than in the previous example.

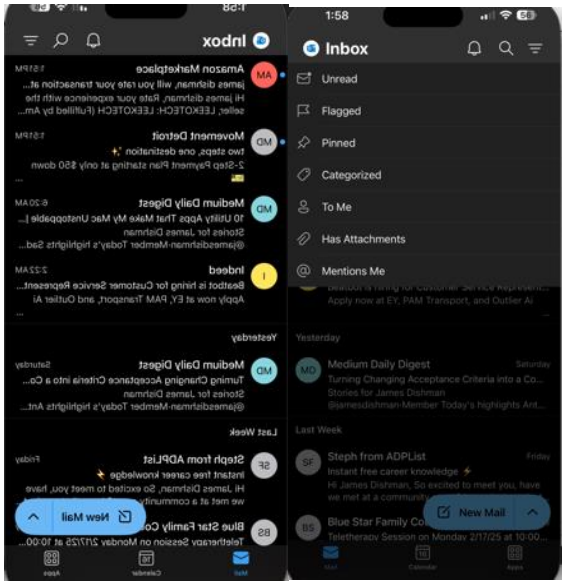

Figure 3 Sliding drawer in Outlook.

The final example of the sliding drawer shows the icon located in a third location, and the icon is different from the other examples. This drawer allows the user to sort the email in their inbox.

b) Upper area tabs (top tabs, top navigation tabs, tab navigation)

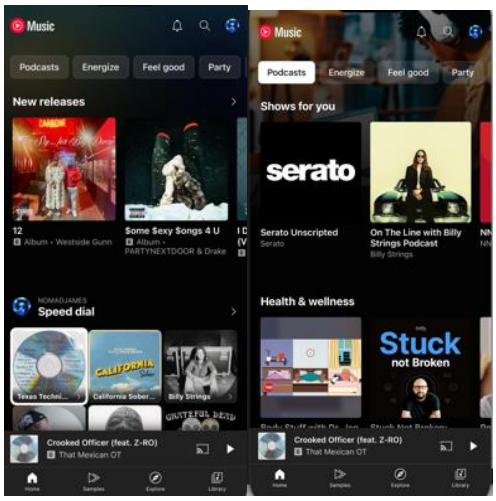

Figure 4 Upper-area tabs in YouTube Music.

YouTube Music utilizes upper-area tabs to allow the user to sort content. The tabs sort the music by type (podcasts) and mood. The tabs are unique because they work as toggles, so when the user deactivates the tab, it returns to the regular view on the home screen. In the first image, if you look closely at the tab on the furthest right, you can see a partial tab indicator signifying additional tabs, which can be accessed by scrolling left.

Figure 5 Upper-area tabs in Discord.

Discord has three upper-area tabs: Search, Add Friend, and Events. These tabs are generally used less frequently than those at the bottom of the application.

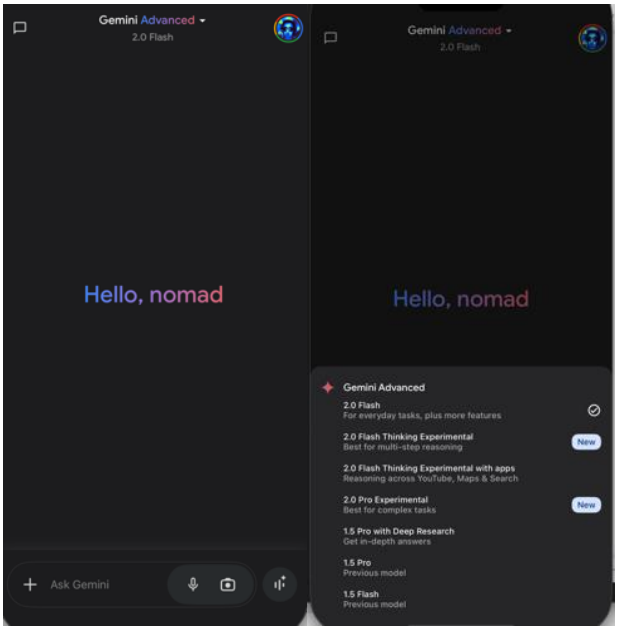

Figure 6 Upper-area tabs in Google Gemini.

Google Gemini also has three upper-area tabs. The first starts a new conversation, the middle allows the user to choose what reasoning model they want to use, and the last allows the user to access their Google profile. Reasoning models are unique to AI applications and significantly influence the application's user experience and function.

c) Bottom menu (bottom navigation, bottom bar, bottom tabs, navigation bar, tab bar)

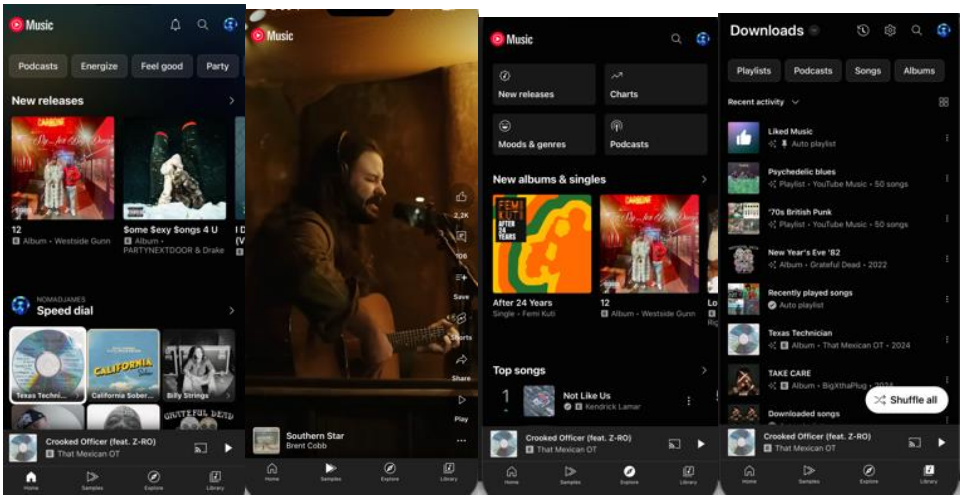

Figure 7 Bottom bar in YouTube Music.

The bottom bar in YouTube Music comprises two parts throughout most of the application. The first part is the device selector and play button, common options amongst media player applications. The second layer comprises Home, Samples, Explore, and Library. Home, Explore, and Library all make sense for a media playing application, while samples seems arbitrary.

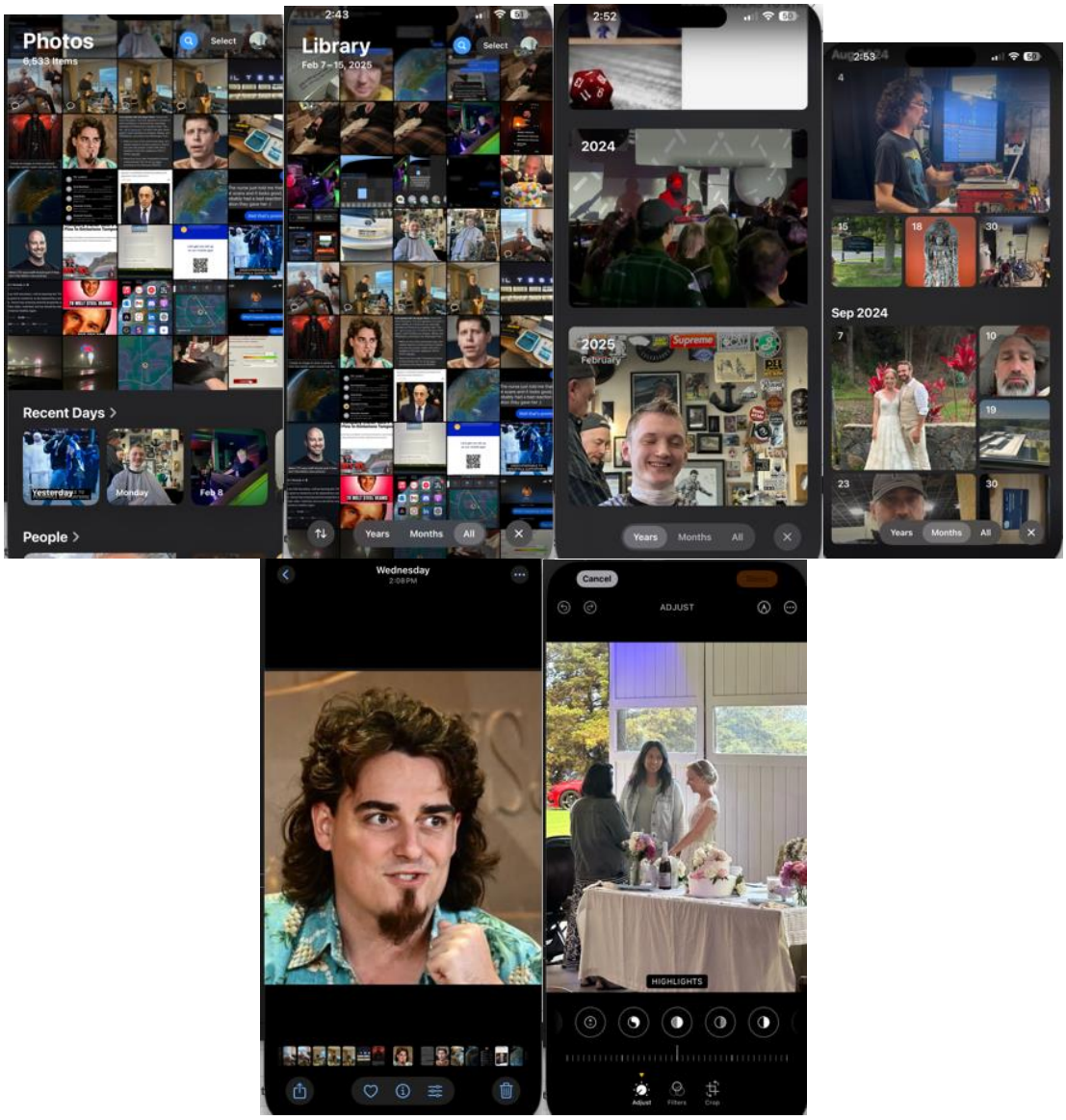

Figure 8 Bottom menus in iOS Photos.

The bottom menus in iOS Photos change depending on the page type and where users interact with the application. When the user is in the default view of the Photos library, the bottom menu is hidden and only comes into view when the user starts to scroll upward through the photo library. It changes again when a photo is selected. If you notice, the image preview changes whenever the user switches between years, months, and all. The last photo shows how the options change again when a photo is selected. The user can share, like, look at the metadata, adjust, and delete the photo. The bottom menu changes again when the user chooses to adjust the photograph. These interactions are complex and are not explained to the user. Still, this selective disclosure makes the Photos application simple enough that the user can understand what is necessary to use it.

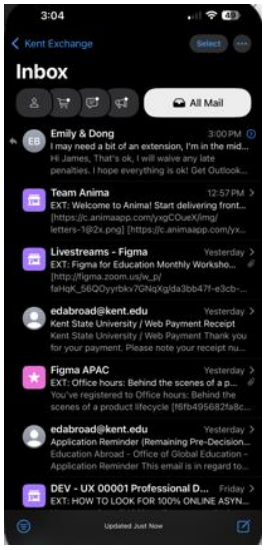

Figure 9 Bottom menu in iOS Mail.

The iOS Mail application has the most uncomplicated bottom menu of these examples. Here, you can change the view between the selected mailbox and all mailboxes and create a new email.

d) Floating Action Buttons (FABs) (Floating button, quick action button)

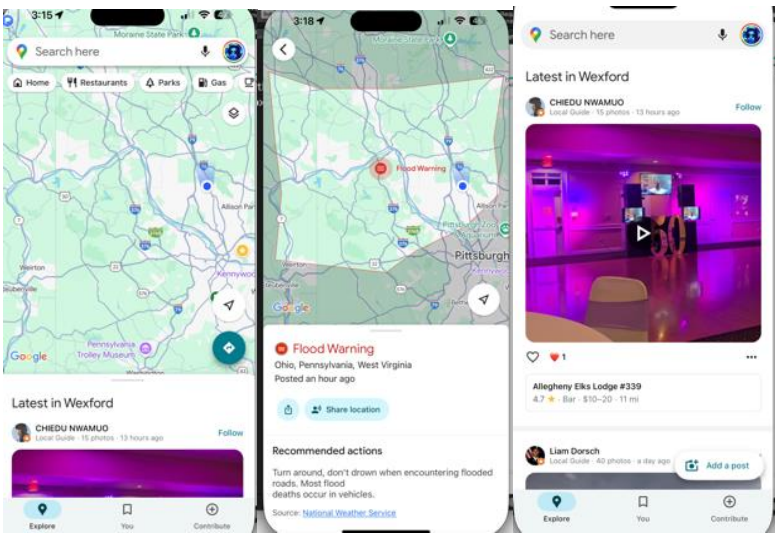

Figure 10 Floating Action Buttons in Google Maps.

Google Maps utilizes floating action buttons, but only when the user is in the Explore section of the application. These buttons change based on the user's actions, as demonstrated in the above picture. These buttons are not moveable, but they allow the user to see more information, which is valuable in a mapping application.

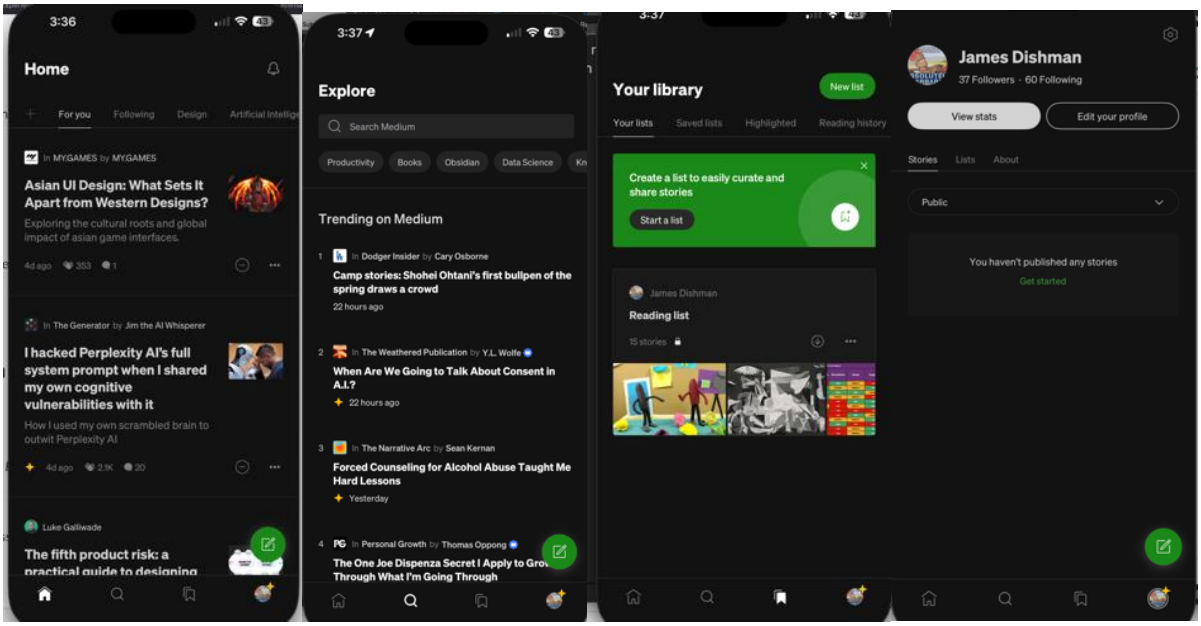

Figure 11 Floating action button in Medium.

Strangely, Medium has one floating action button, and its purpose is to publish. Even stranger, it shows up on every page except for Your Library. This design choice is not consistent and illogical. Most Medium users are most likely using Medium as a reader and are not publishing, so why is the task given special preference?

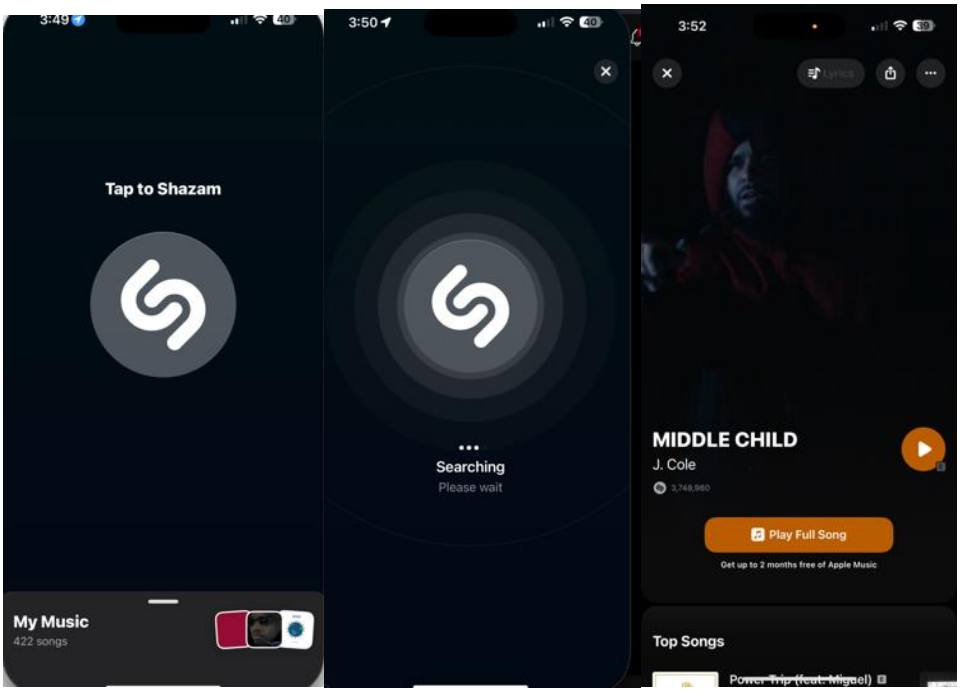

Figure 12 Floating action button in Shazam.

Another odd example of a floating action button is the primary action button in the Shazam application, a music recognition tool. One large button is floating over the entire screen with a singular purpose: to analyze and identify music. It is strange because it dominates the whole application and doesn't actually float over any information. Still, it is helpful because the application has one primary purpose: identifying music, usually in a loud and dark environment.

2) Content Layouts

a) Filmstrip

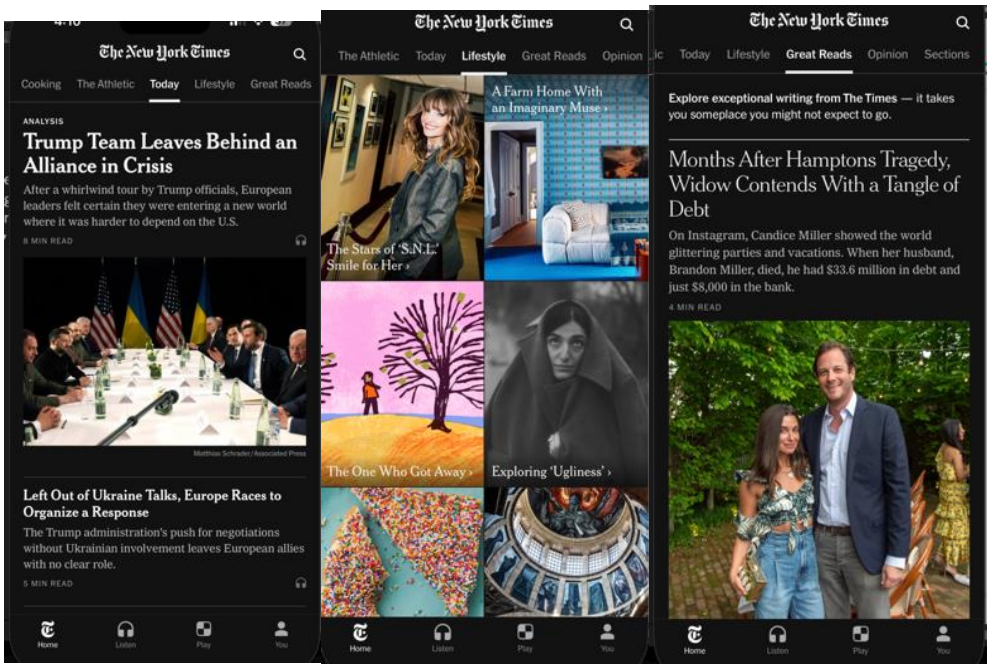

Figure 13 Filmstrip layout in NYTimes application.

The NYTimes application utilizes the filmstrip layout to differentiate between sections. It also uses different layouts within the sections themselves, including cards, collections, and infinite lists, which is interesting.

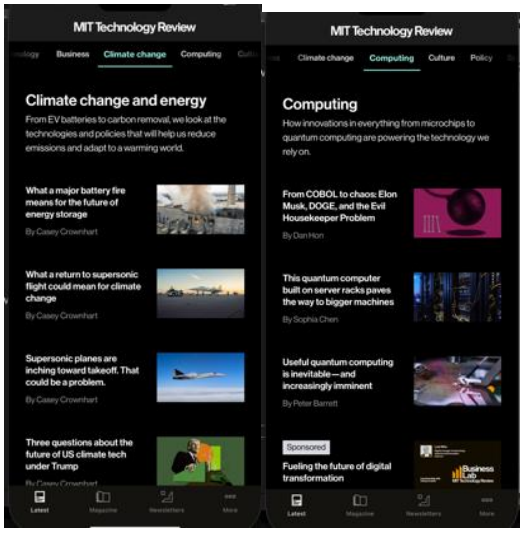

Figure 14 Filmstrip layout in the MIT Technology Review application.

The MIT Technology Review application uses a similar format to distinguish between sections. Their layout is interesting because it is simultaneously a filmstrip layout and an upper-area tab. Like the NYTimes application, the section's content is presented as a card.

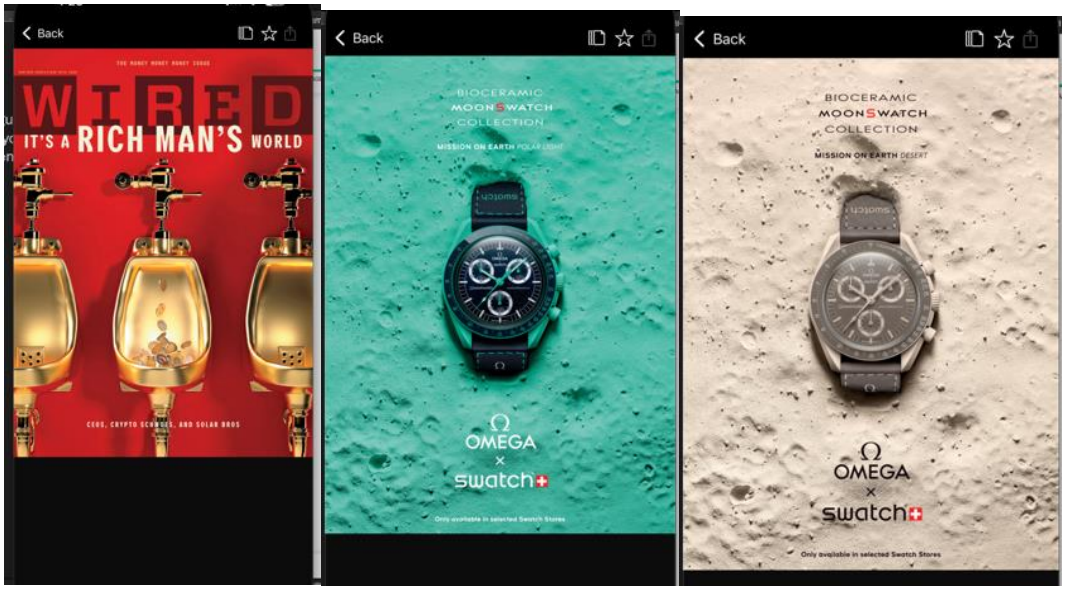

Figure 15 Filmstrip layout in the Wired application.

The Wired application utilizes a straightforward filmstrip layout whenever users are inside an issue of the magazine. The user can swipe left or right to maneuver through the pages like a physical magazine. This is not a particularly usable solution for looking at a magazine on the screen of a

phone, as the user will routinely have to zoom in to read the content and then zoom out to navigate through the issue.

b) Collections and Cards (card layout)

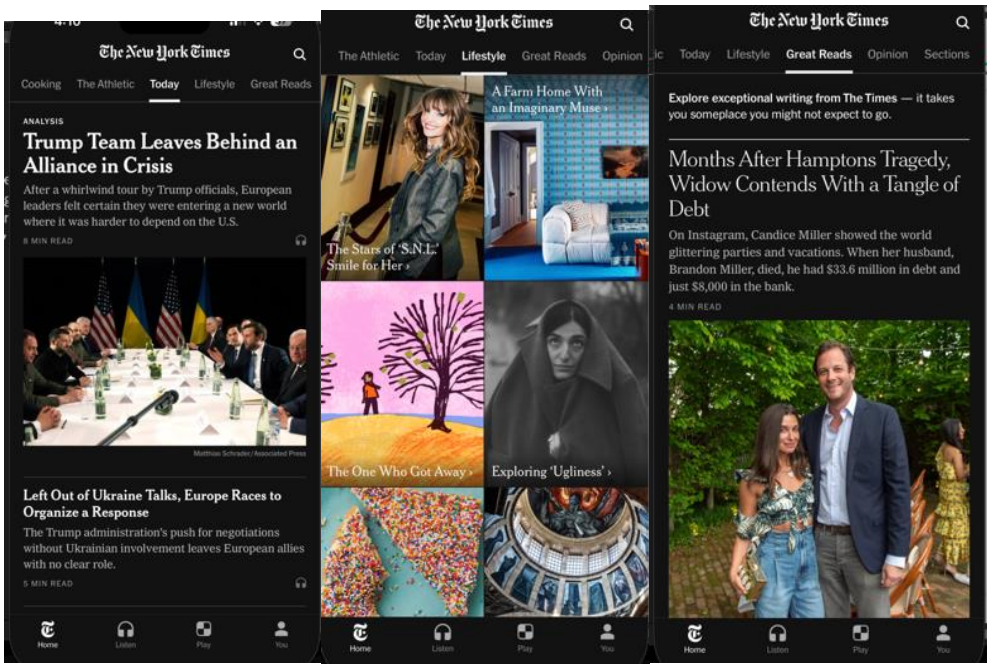

Figure 16 Collections and cards in the NYTimes Application.

The NYTimes application utilizes a filmstrip layout to differentiate between sections. Still, it utilizes collections and cards to distinguish between the articles in the sections while using an infinite list to populate them. This delivery system is well thought out and works well to deliver the voluminous content of the New York Times.

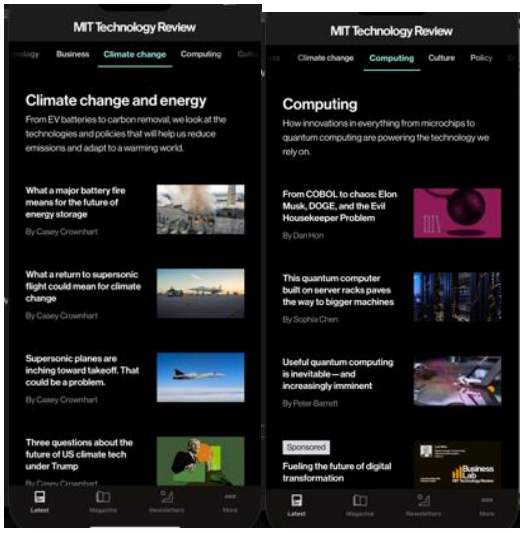

Figure 17 Cards in the MIT Technology Review application.

The MIT Technology Review application employs a similar mixed delivery method as the NYTimes application, possibly in a less eloquent form. However, it works well to differentiate between sections and articles.

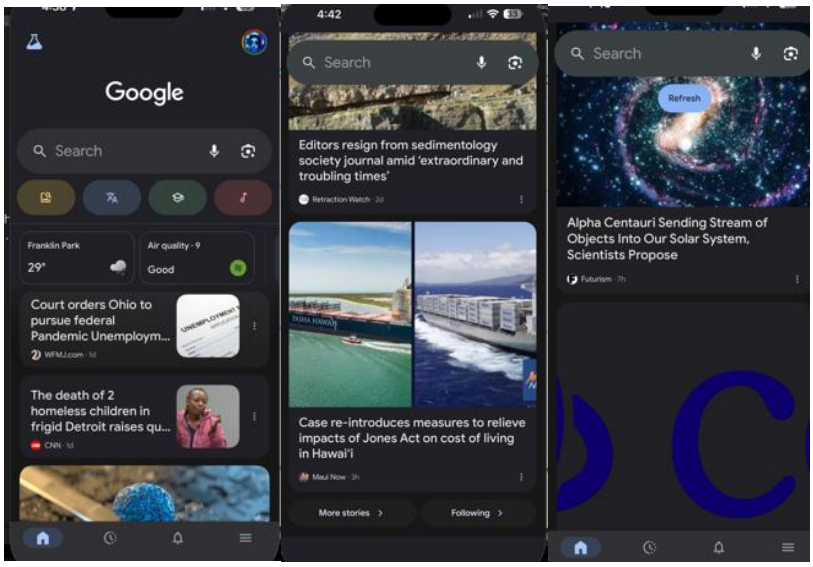

Figure 18 Cards in the Google application.

The Google application uses cards to deliver articles to the user. Using a thumbnail and text lead increases engagement and allows the user to preview the content before committing to reading it. This application also

utilizes the version of infinite list, where the user can choose to load More Stories or to hit the floating Refresh button to load additional articles.

c) Infinite list (infinite scrolling, continuous scrolling, lazy loading list)

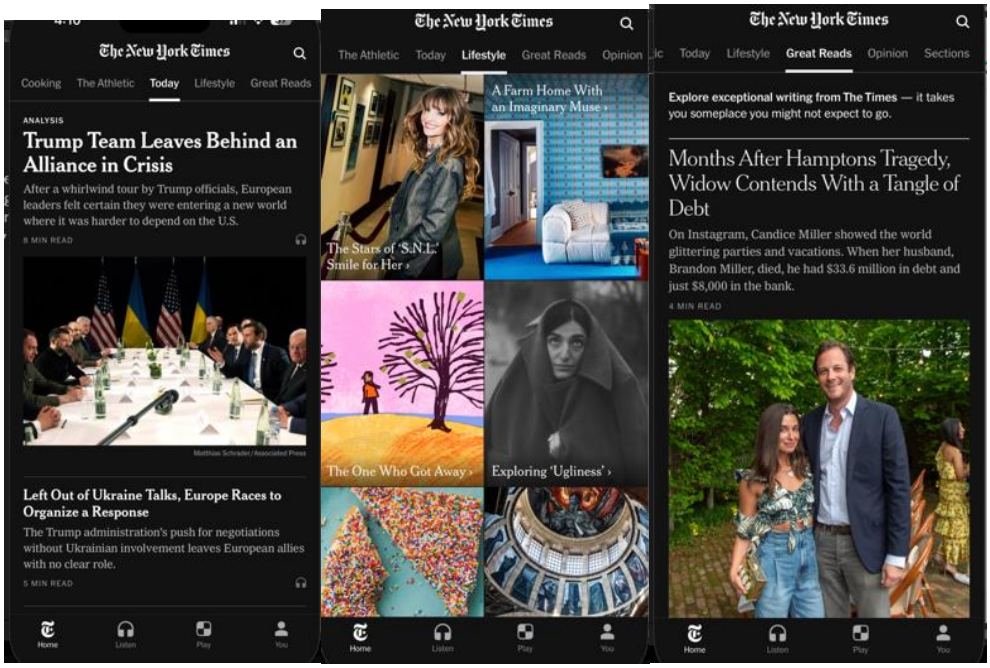

Figure 19 Infinite list in the NYTimes application.

We return to the NYTimes application one last time. While this application utilizes filmstrip, collections, and cards, it is also algorithmically populated and continues to load content as the user scrolls down through the articles. It is an interesting hybrid of content delivery technologies.

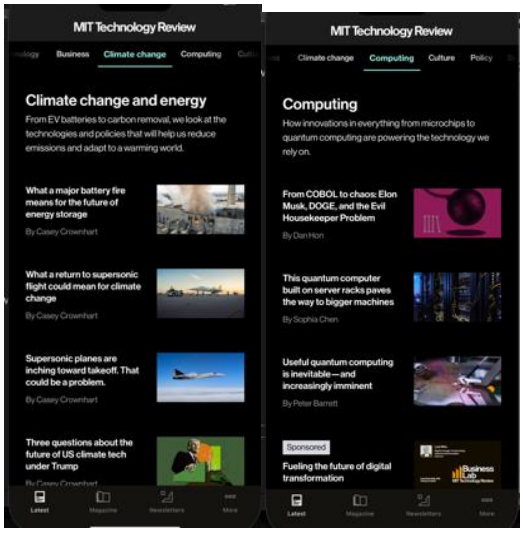

Figure 20 Infinite list in the MIT Technology Review application.

Yet again, the MIT Technology Review application uses a mixed delivery system like the NY Times application. The user can infinitely scroll downward through articles published in the application.

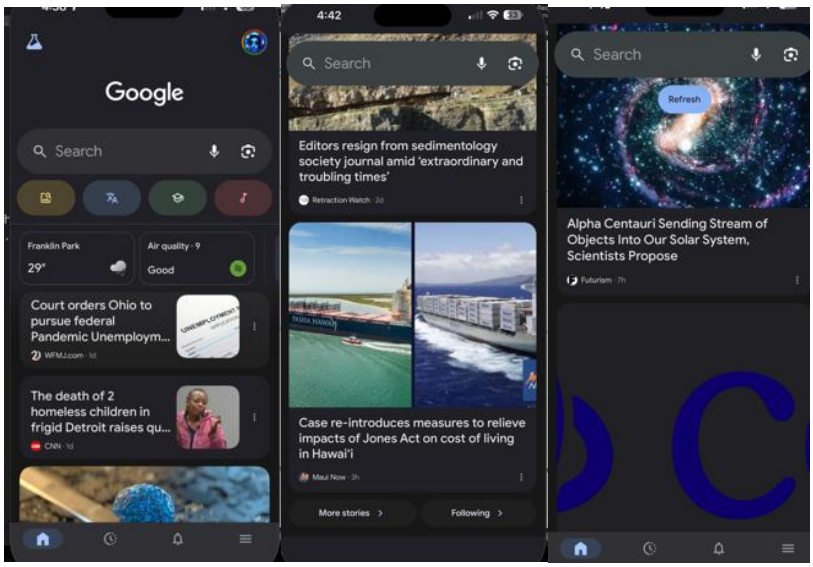

Figure 21 Infinite list in the Google application.

For the sake of consistency, we can look at how Google populates new articles for the user, it also uses an infinite list, the difference is that user needs to choose either More Stories or Refresh to load additional content.