



HOMEWORK 9:

MOBILE APP DEVELOPMENT – iOS

Prof. Marco Papa

Designed and Developed By – Prachi Tawalare

CSCI 571: Web Technologies

Homework 9: iOS News App

Table of Contents

1. Objectives.....	2
2. Background	2
2.1 Xcode	2
2.2 iOS	2
2.3 Swift.....	3
3. Prerequisites.....	3
3.1 Download and install Xcode	3
3.2 Add your account to Xcode	3
3.3 Install CocoaPods.....	4
4. High Level Design.....	4
5. Implementation	6
5.1 Home Tab.....	6
5.1.1 First Sub View	6
5.1.3 Home TableView	7
5.1.4 Pull Down Refresh	9
5.1.5 Context Menu.....	9
5.1.6 Detailed Article Page	10
5.1.7 Search Controller.....	12
5.1.8 Search Results Page.....	13
5.2 Headlines Tab.....	15
5.1.1 XLPagerTabStrip	15
5.3 Trending Tab	18
5.4 Bookmarks Tab.....	19
5.4.1 Adding to Bookmark.....	19
5.4.2 Removing from Bookmark.....	20
5.4.3 Empty Bookmark Tab	21
5.4.4 Bookmarks Implemetation	22
6. Twitter Share	22
7. Additional Information.....	23
8. Hints.....	23
9. Images / Icons	23
10. Material You Need to Submit	23

1. Objectives

- Become familiar with the Swift language, Xcode and iOS App development.
- Practice the Model-View-Controller design pattern.
- Build a good-looking iOS app.
- Learn to integrate APIs and iOS SDK
- Manage and use third-party libraries by CocoaPods

2. Background

2.1 Xcode

Xcode is an integrated development environment (IDE) containing a suite of software development tools developed by Apple for developing software for macOS and iOS. First released in 2003, the latest stable release is version 11.1 and is available via the Mac App Store free of charge.

Features:

- Swift 5 support
- Playgrounds
- Interface Builder
- Device simulator and testing
- User Interface Testing
- Code Coverage

The Official homepage of the Xcode is located at:

<https://developer.apple.com/xcode/>

2.2 iOS

iOS (originally iPhone OS) is a mobile operating system created and developed by Apple Inc. and distributed exclusively for Apple hardware. It is the operating system that presently powers many of the company's mobile devices, including the iPhone, iPad, and iPod touch. It is the second most popular mobile operating system in the world by sales, after Android.

The Official iOS home page is located at:

<http://www.apple.com/iOS/>

The Official iOS Developer homepage is located at:

<https://developer.apple.com/ios/>

2.3 Swift

Swift is a general-purpose, multi-paradigm, compiled programming language created for iOS, macOS, watchOS, tvOS and Linux development by Apple Inc. Swift is designed to work with Apple's Cocoa and Cocoa Touch frameworks and the large body of existing Objective-C code written for Apple products. Swift is intended to be more resilient to erroneous code ("safer") than Objective-C and also more concise. It is built with the LLVM compiler framework included in Xcode 6 and later and uses the Objective-C runtime, which allows C, Objective-C, C++ and Swift code to run within a single program.

The Official Swift homepage is located at:

<https://developer.apple.com/swift/>

3. Prerequisites

This homework requires the use of the following components:

3.1 Download and install Xcode

To develop iOS apps using the latest technologies described in these lessons, you need a Mac computer (macOS Sierra 10.12.4 or later) running the Xcode (11.3). Xcode includes all the features you need to design, develop, and debug an app. Xcode also contains the iOS SDK, which extends Xcode to include the tools, compilers, and frameworks you need specifically for iOS development.

For HW-9, Xcode version 11.3 can be used.

Xcode version 11.3 is strongly recommended. It can be downloaded from the following link:

<https://developer.apple.com/download/more/?name=Xcode>

You may use any other IDE other than Xcode, but you will be on your own if problems come up.

Swift 5.1 is strongly recommended to use for this app.

3.2 Add your account to Xcode

When you add your Apple ID to the Xcode Accounts preferences, Xcode displays all the teams you belong to. Xcode also shows your role on the team and details about your signing identities and provisioning profiles that you'll create later in this document. If you don't belong to the Apple Developer Program, a personal team appears.

Here is detailed documentation:

<https://developer.apple.com/library/iOS/documentation/IDEs/Conceptual/AppStoreDistributionTutorial/AddingYourAccounttoXcode/AddingYourAccounttoXcode.html>

3.3 Install CocoaPods

CocoaPods is a dependency manager for Swift and Objective-C Cocoa projects. It has over ten thousand libraries and can help you scale your projects elegantly. You can install dependencies using it, we will need to install many third-party modules and frameworks using it.

CocoaPods is built with Ruby and is installable with the default Ruby available on macOS. We recommend you use the default Ruby. Using the default Ruby install can require you to use 'sudo' when installing gems.

Run the command below in your Mac terminal:

```
$ sudo gem install cocoapods
```

Once you have created your Xcode project, you can start to integrate CocoaPods into your project.

Further guides on how to integrate CocoaPods are available at: <https://cocoapods.org/>.

The following pods will be needed:

- [SwiftJSON](#): To handle JSON data.
- [Alamofire](#): ~> 4.9.1: To make HTTP requests. (Use correct version to avoid compatibility Issues)
- [Toast-Swift](#) ~> 4.0.0* : To display toast messages in your app
- [SwiftSpinner](#) : For the big loading spinner.
- [Charts](#): For implementing charts on Trending Tab.
- [XLPagerTabStrip](#) ~> 9.0: For implementing the Headlines Tab

4. High Level Design

This homework is a mobile app version of Homework 8. In this exercise, you will develop an iOS Mobile application, which allows users to search and display news articles using the guardian news APIs, add and remove articles from bookmarks, display a chart indicating the interest over time for a particular term and Share articles using Twitter API. You can reuse the backend service (node.js script) you developed in HW8 and follow the same API call requirements.

The first scene of this app is shown in Figure 1.3. The launch screen of the App is shown in Figure 1.1. The initial View is a [UITabBarController](#) with four tabs:

1. Home
2. Headlines
3. Trending
4. Bookmarks

All the implementation details and requirements will be explained in the following sections. The following App Icon should be used for the App. It is provided in the Images/Icons section.





Figure 1.1: Launch Screen

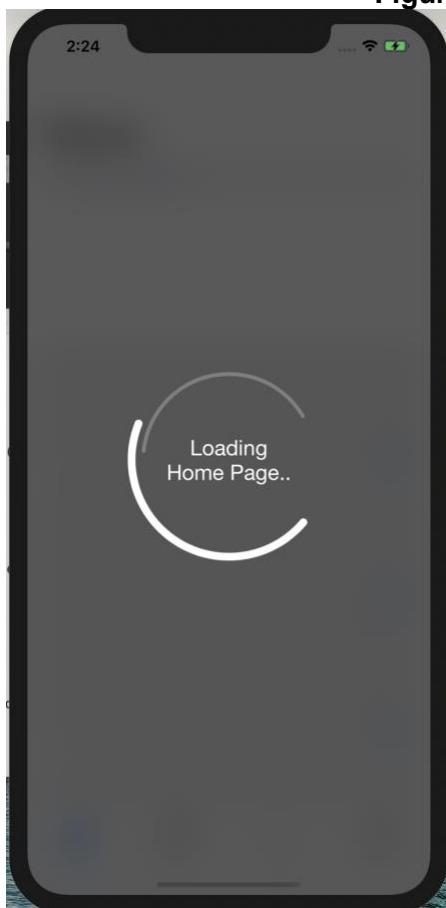


Figure 1.2: Display a Loading spinner while displaying Home Tab

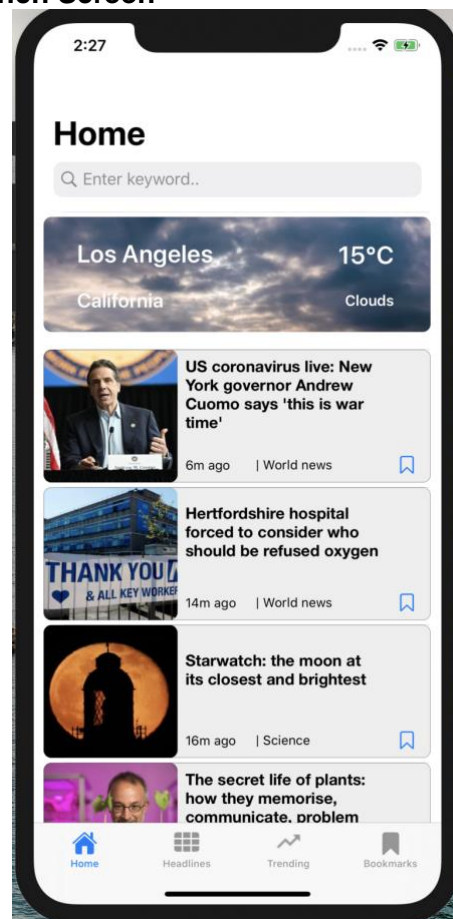


Figure 1.3: Home Tab

5. Implementation

5.1 Home Tab

You must replicate the Initial View as shown in Figure 1.3. It should **always** display the weather details of the Current location automatically.

The interface consists of the following:

5.1.1 First Sub View

The first subview consists of temperature, weather summary and the City and State Name (Figure 3) of the current user location. The background image for the weather card will change according to the value of the weather summary (Figure 3.3).

To obtain the current location, standard location service provided by apple should be used. You have to allow the app to access the device location.

More details on standard location service can be found here:

https://developer.apple.com/documentation/corelocation/getting_the_user_s_location/using_the_standard_location_service

After Obtaining the location co-ordinates you need to convert it to user-friendly place names(**City and State names**) as described in the link below:

https://developer.apple.com/documentation/corelocation/converting_between_coordinates_and_user-friendly_place_names

After this you will need the city name to get the weather details using the Openweathermap API.

Note: For getting the weather information, the Openweathermap API call has be made directly from iOS mobile client (Swift) to the Web Service (Not from Node.js backend).

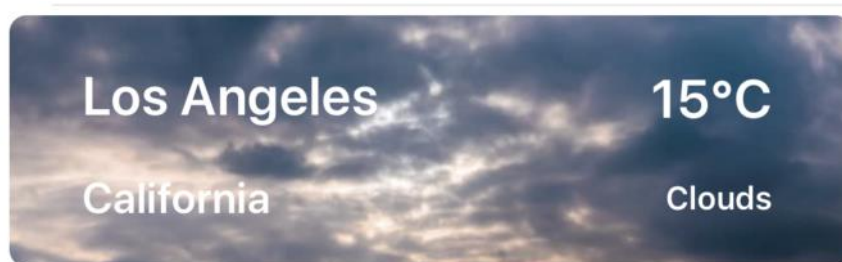


Figure 2: First Sub View

Openweathermap API Call

[https://api.openweathermap.org/data/2.5/weather?q=\[CITY\]&units=metric&appid=\[API_KEY\]](https://api.openweathermap.org/data/2.5/weather?q=[CITY]&units=metric&appid=[API_KEY])

The response would be as shown in Figure 3.1:

```

▼ coord:
  lon:      -118.24
  lat:      34.05
▼ weather:
  ▼ 0:
    id:      800
    main:    "Clear"
    description: "clear sky"
    icon:    "01d"
    base:    "stations"
  ▼ main:
    temp:    18.71
    feels_like: 16.72
    temp_min: 15.56
    temp_max: 21.11
    pressure: 1014
    humidity: 49
    visibility: 14484
  ▶ wind:    {...}
  ▶ clouds:  {...}
  dt:      1585942764
  ▶ sys:    {...}
    timezone: -25200
    id:      5368361
    name:    "Los Angeles"
    cod:    200

```

Figure 3.1: JSON response from OpenWeatherAPI

Fields in the view	Corresponding fields in JSON
Temperature (in Celsius)	main ~> temp (rounded to nearest integer)
Summary	weather[0] ~> main

Figure 3.2: Fields in the view to JSON data mapping for OpenWeather API

Summary	Image
Clouds	cloudy_weather
Clear	clear_weather
Snow	snowy_weather
Rain	rainy_weather
Thunderstorm	thunder_weather
Default(For all other values)	sunny_weather

Figure 3.3: Summary value to image mapping

5.1.3 Home TableView

Home Tab tableview is a scrollable [UITableView](#), with each cell displaying the data for the news articles retrieved from the Guardian API. The data consists of the article Title, article Image, the time since the article was published, category of the article and a button to bookmark that article. You should also add a pull-down refresh feature for this table which will fetch the latest news article and reload the table and pull fresh data from the backend.

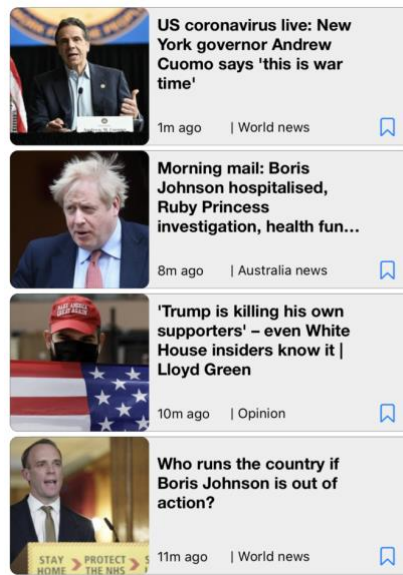


Figure 4: Home TableView

The HomePageTableView provides a list of 10 latest news articles from the Guardian API. To API call to get the latest news is as follows:

[https://content.guardianapis.com/search?orderby=newest&showfields=starRating,headline,thumbnail,short-url&api-key=\[YOUR_API_KEY\]](https://content.guardianapis.com/search?orderby=newest&showfields=starRating,headline,thumbnail,short-url&api-key=[YOUR_API_KEY])

The response from the Guardian API is shown in Figure 5:

```

▼ response:
  status: "ok"
  userTier: "developer"
  total: 2183450
  startIndex: 1
  pageSize: 10
  currentPage: 1
  pages: 218345
  orderBy: "newest"
▼ results:
  ▼ 0:
    id: "football/2020/apr/04/liverpool-become-latest-premier-league-club-to-furlough-non-playing-staff"
    type: "article"
    sectionId: "football"
    sectionName: "Football"
    webPublicationDate: "2020-04-04T19:56:49Z"
    webTitle: "Liverpool under fire for furloughs while PFA points to pay-cut tax trap"
    webUrl: "https://www.theguardian.com/football/2020/apr/04/liverpool-become-latest-premier-league-club-to-furlough-non-playing-staff"
    apiUrl: "https://content.guardianapis.com/football/2020/apr/04/liverpool-become-latest-premier-league-club-to-furlough-non-playing-staff"
    fields:
      headline: "Liverpool under fire for furloughs while PFA points to pay-cut tax trap"
      shortUrl: "https://gu.com/p/dted5"
      thumbnail: "https://media.guim.co.uk/120210a9c1a3d290be3e71f5f8bf9230f782807e/0_186_6720_4032/500..jpg"
      isHosted: false
      pillarId: "pillar/sport"
      pillarName: "Sport"
    ▶ 1: {}
    ▶ 2: {}
    ▶ 3: {}
    ▶ 4: {}

```

Figure 5: Response JSON for Home Tab

Each News card has the following properties:

- **Image:** “thumbnail” property of the “fields” in the “results” array of the “response” json.
- **Title:** “webTitle” property in the “results” array of the “response” json. Need to show ellipsis after 3 lines. Breaking words is okay.
- **Time:** “webPublicationDate” property in the “results” array of the “response” json.
- **Section:** “sectionName” property in the “results” array of the “response” json.
- **Bookmark Icon:** On clicking the bookmark icon, that particular news will be bookmarked, and the bookmark icon will change. Similarly, if the news is already bookmarked, it will be removed from bookmarks and the bookmark icon will change back. Xcode 11 provides these Icons.
- **Article ID:** “id” property in the “results” array of the “response” json. This will be used for the API call to fetch details of each article (Refer section 5.6)

For the Time, webPublicationDate and get the difference in time from the Current date and display them as:

“h ago” if the difference is greater than 1 hour. (e.g. 12h ago).

“m ago” if the difference is greater than 1 minute but less than 1 hour. (e.g. 12m ago).

“s ago” if the difference is less than a minute. (e.g. 12s ago)

If any image is not available use the default Guardians Image Provided.

5.1.4 Pull Down Refresh

The Pull Down Refresh feature should be added to tables on Home and Headlines tab and also on the Search Results page.

5.1.5 Context Menu

Long Press on the table Cells of Home and Headlines Tab and the Search Results Page should open a [Context Menu](#) with two Menu items as shown in Figure 6 below. (see Hints section)

1. Share with twitter
2. Bookmark

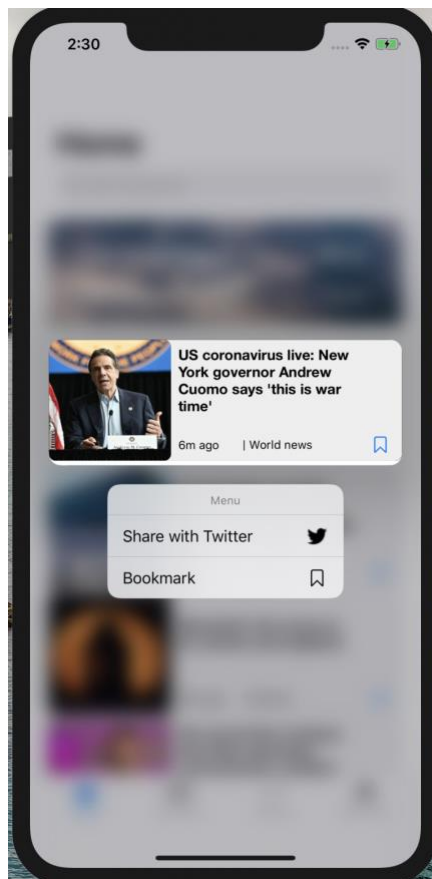


Figure 6: Home Page Context Menu

5.1.6 Detailed Article Page

Selecting the articles from Home Tab, Headlines Tab or Search Tab opens the Detailed Article Page as shown in Figure 6. It consists of the article Image, Title, Description, Date Published, Category and a “View Full Article” hyperlink. All the contents on the page are scrollable and to implement this you should use the [UIScrollView](#) (See Hints). In addition, pay attention to the navigationBar of this Page. It has the back button which takes you back to the previous scene, the article title and bookmark and twitter Buttons. You should make use of [UINavigationController](#) to achieve this as shown in Figure 7.

To get the details of the article, call the following API:

[https://content.guardianapis.com/\[ARTICLE_ID\]?api-key=\[YOUR_API_KEY\]&show-blocks=all](https://content.guardianapis.com/[ARTICLE_ID]?api-key=[YOUR_API_KEY]&show-blocks=all)

The **ARTICLE_ID** is fetched from the JSON response shown in Figure 6 by accessing the “id” property in the “results” array of the “response” json.

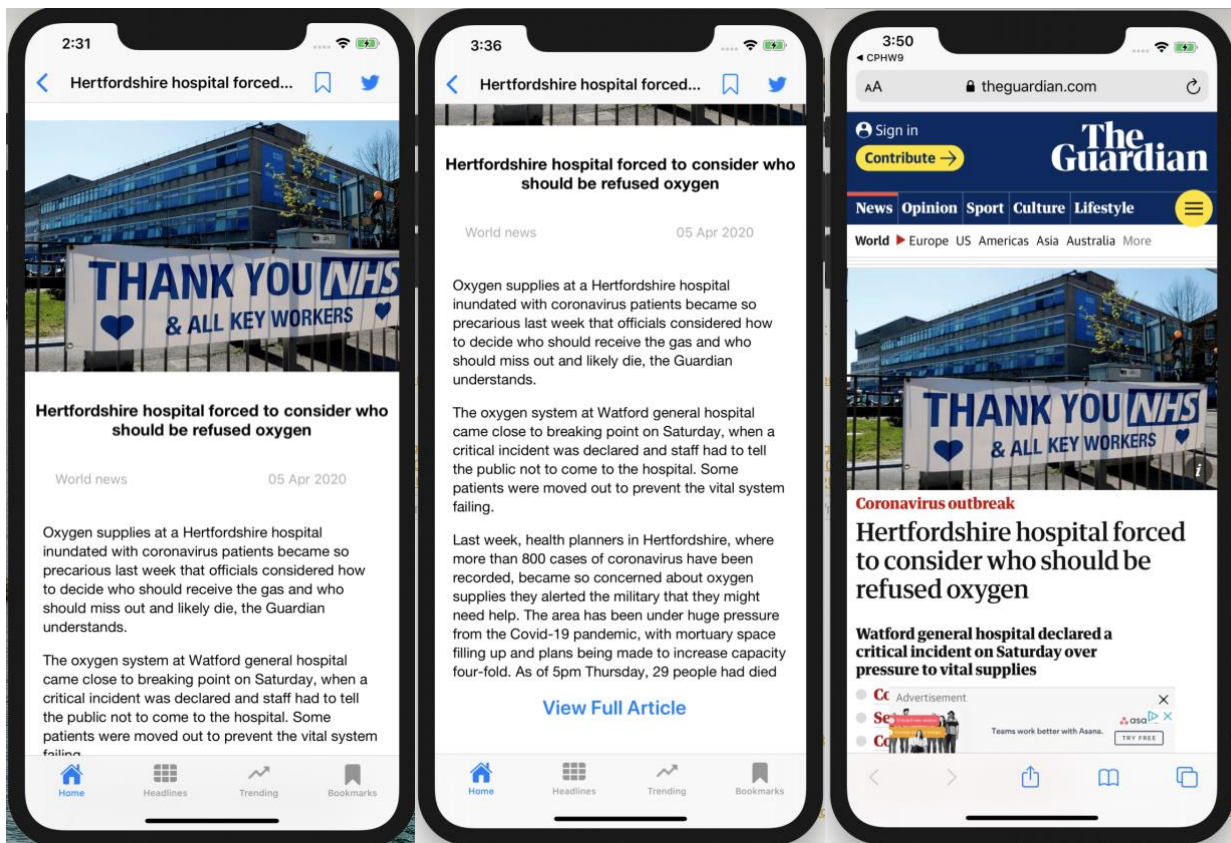


Figure 7: Detailed Article Pages

```

{
  "response": {
    "status": "ok",
    "userTier": "developer",
    "total": 1,
    "content": {
      "id": "us-news/2020/feb/03/trump-impeachment-trial-closing-arguments",
      "type": "article",
      "sectionId": "us-news",
      "sectionName": "US news",
      "webPublicationDate": "2020-02-03T20:57:00Z",
      "webTitle": "Trump impeachment trial: Democrats warn Trump 'will do it again' if acquitted",
      "webUrl": "https://www.theguardian.com/us-news/2020/feb/03/trump-impeachment-trial-closing-arguments",
      "apiUrl": "https://content.guardianapis.com/us-news/2020/feb/03/trump-impeachment-trial-closing-arguments",
      "fields": {
        "headline": "Trump impeachment trial: Democrats warn Trump 'will do it again' if acquitted",
        "shortUrl": "https://gu.com/p/d7nag",
        "thumbnail": "https://media.guim.co.uk/22b0ced6c17120f831eb237ae8ca524cc27bb903/0_21_1722_1033/500.jpg"
      },
      "blocks": {
        "main": {
          "body": {
            "id": "5e38371a8f086a28115e4b13",
            "bodyHtml": "<p>Warning that 'history will not be kind to Donald Trump,' the Democratic representative Adam Schiff mounted an impassioned closing argument in the Senate impeachment trial on Monday, urging the chamber to hold the president to account.</p><aside class='element element-rich-link element--thumbnail'><p><span>Related:</span></p><href>'https://www.theguardian.com/us-news/2020/feb/02/republican-joe-walsh-iowa-trump'>My party is a cult': Republican Joe Walsh on his Iowa challenge to Trump</a></p></aside><p>The House impeachment managers, who are prosecuting Trump,<strong></strong>pleaded with Senate Republicans to find Trump guilty of the charges in the two articles of impeachment: abuse of power and obstruction of Congress.</p><p>'History will not be kind to Donald Trump,' Schiff said. 'I think we all know that. And if you find that the House has proved his case and still vote to acquit, your name will be tied to his with a cord of steel for all of history.</p><p>Schiff blasted Trump in personal terms, warning that Trump had tried to cheat in the 2020 election and will keep trying if acquitted.</p><p>'He has not changed. He will not change,' said Schiff. 'A man without character or ethical compass will never find his way. He has done it before and he will do it again. What are the odds if he is left in office that he will continue to try to cheat? I will tell you: 100%.</p><p>'He will continue to try to cheat in the election until he succeeds. Then what shall you say?'</p><p>But a mostly party-line vote on Friday not to call witnesses in the trial signaled <a href='https://www.theguardian.com/us-news/2020/feb/02/republican-joe-walsh-iowa-trump'>a lack of interest on the Republican side.</p><p>Trump maintains that his conduct has been irreproachable, and all but a handful of the 53 senators in the majority have seamlessly agreed.</p><p>'This is an effort to overturn the results of one election and to try to interfere in the coming election that begins today in Iowa,' White House counsel Pat Cipollone, who is leading Trump's impeachment defense, told the chamber.</p><p>'The only appropriate result here is to acquit the president and to leave it to the voters to choose their president.</p><p>The closing arguments set the stage for a vote to acquit Trump on Wednesday afternoon. The senators will hold separate votes on each article of impeachment, with an out-of-reach two-thirds majority needed to convict Trump and remove him from office.</p><p>'The third president in US history to be impeached would then become the third president also

```

Figure 8: Detailed Article JSON Response

Each detailed article page (shown in Figure 8) has the following properties:

- **Image:** “file” property of the “assets” array in the “elements” array in the “main” object in the “blocks” object in the “results” array of the “response” json.

- **Title:** “webTitle” property in the “content” object of the “response” json.
- **Date:** “webPublicationDate” property in the “content” object of the “response” json.
- **Section:** “sectionName” property in the “content” object of the “response” json.
- **Description:** Append every “bodyHtml” property in the “body” array of the “blocks” object in the “content” object of the “response” json.
- **Article URL (for View Full Article at the bottom):** “webUrl” property in the “content” object of the “response” json.

For the Date, convert the webPublicationDate in the format “dd MMM yyyy”.

Notes:

- If any image is not available, use the default Guardian image provided.
- The “View Full Article” is clickable and redirects to the original article on Guardian news website.
- Limit the description to 30 lines and add ellipsis at the end. Breaking word is okay.

5.1.7 Search Controller

A [UISearchController](#) component of the “navigationItem” allowing the user to enter the **keyword** that the user wants to search. While typing in the search bar, the results should be provided using the Bing autosuggest API. Make sure you use the same API as Homework 8. (Hint: The results can be shown as a ‘UITableView’ which is hidden/shown according to the response of the autosuggest API).

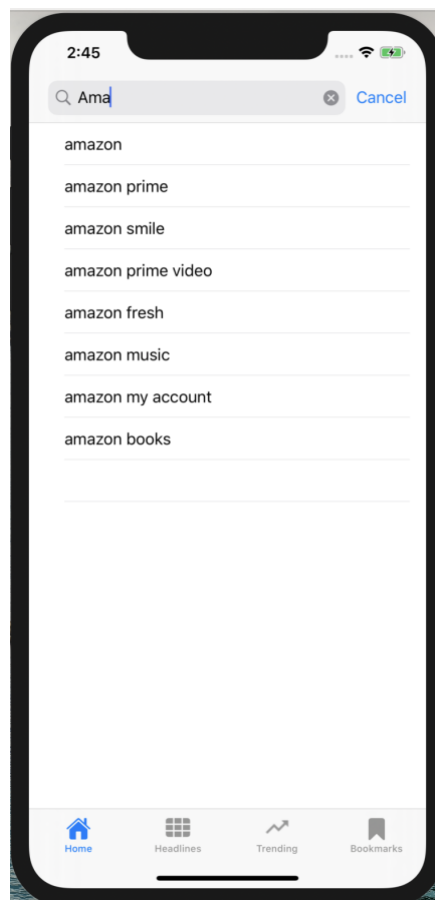


Figure 9: AutoSuggest for Search

For the autosuggest API, call the following API directly from your iOS App (No need to use Node.js):

[https://api.cognitive.microsoft.com/bing/v7.0/suggestions?q=\[KEYWORD\]](https://api.cognitive.microsoft.com/bing/v7.0/suggestions?q=[KEYWORD])

Similar to that in Homework 8, you will have to pass the 'Ocp-Apim-Subscription-Key' to make the API request. The response from the API is shown in Figure 10:

For the suggestions, use the "displayText" property of the "searchSuggestions" array in the 0th index object of the "suggestionGroups" array.

```
{
  "_type": "Suggestions",
  "queryContext": {
    "originalQuery": "ama"
  },
  "suggestionGroups": [
    {
      "name": "Web",
      "searchSuggestions": [
        {
          "url": "https://www.bing.com/search?q=amazon.com&FORM=USBAPI",
          "displayText": "amazon.com",
          "query": "amazon.com",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=amazon+prime&FORM=USBAPI",
          "displayText": "amazon prime",
          "query": "amazon prime",
          "searchKind": "WebSearch"
        }
      ]
    }
  ]
}
```

Figure 10: Bing Autosuggest API response

5.1.8 Search Results Page

When the user taps any of the keyword from the autosuggest results, your app should display a *big spinner* (Figure 12) before it's ready to show the search results page. Then after it gets data from your backend, it should hide the spinner and display the result page with exactly the same layout as the Home tab Table View, shown in Figure 13.

To fetch the articles based on keyword, call the following API:

[https://content.guardianapis.com/search?q=\[KEYWORD\]&api-key=\[YOUR_API_KEY\]&show-blocks=all](https://content.guardianapis.com/search?q=[KEYWORD]&api-key=[YOUR_API_KEY]&show-blocks=all)

The response from the API is shown in Figure 11:


```

▼ response:
  status: "ok"
  userTier: "developer"
  total: 47663
  startIndex: 1
  pageSize: 10
  currentPage: 1
  pages: 4767
  orderBy: "relevance"
▼ results:
  ▼ 0:
    ▼ id: "us-news/2020/feb/03/trump-impeachment-trial-closing-arguments"
    type: "article"
    sectionId: "us-news"
    sectionName: "US news"
    webPublicationDate: "2020-02-03T20:57:00Z"
    ▼ webTitle: "Trump impeachment trial: Democrats warn Trump 'will do it again' if acquitted"
    ▼ webUrl: "https://www.theguardian.com/us-news/2020/feb/03/trump-impeachment-trial-closing-arguments"
    ▼ apiUrl: "https://content.guardianapis.com/us-news/2020/feb/03/trump-impeachment-trial-closing-arguments"
    ▼ fields:
      ▼ headline: "Trump impeachment trial: Democrats warn Trump 'will do it again' if acquitted"
      shortUrl: "https://gu.com/p/d7nag"
      ▼ thumbnail: "https://media.guim.co.uk/22bbced6c17128f031eb237ae0ca524cc27bb903/0_21_1722_1033/500.jpg"
      isHosted: false
      pillarId: "pillar/news"
      pillarName: "News"
    ▶ 1: {}
    ▶ 2: {}
    ▶ 3: {}
    ▶ 4: {}
    ▶ 5: {}

```

Figure 11: Response JSON for Search Results Page

- All the News Cells on this page have the same properties as the Home Tab TableView.
- If you still have to change Node backend, make sure you don't break the React app (if grading is still in progress) or deploy a newer instance of Node.
- In the Autosuggest, only make an API call to the Bing Autosuggest after the user enters 3 characters. Example: "am" should not display any suggestions but "ama" should have suggestions. Refer video for better understanding.

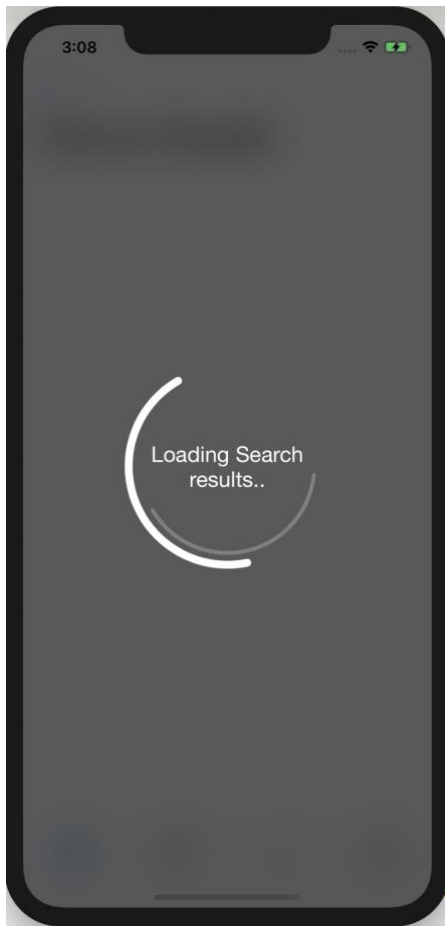


Figure 12: Display a big spinner while searching

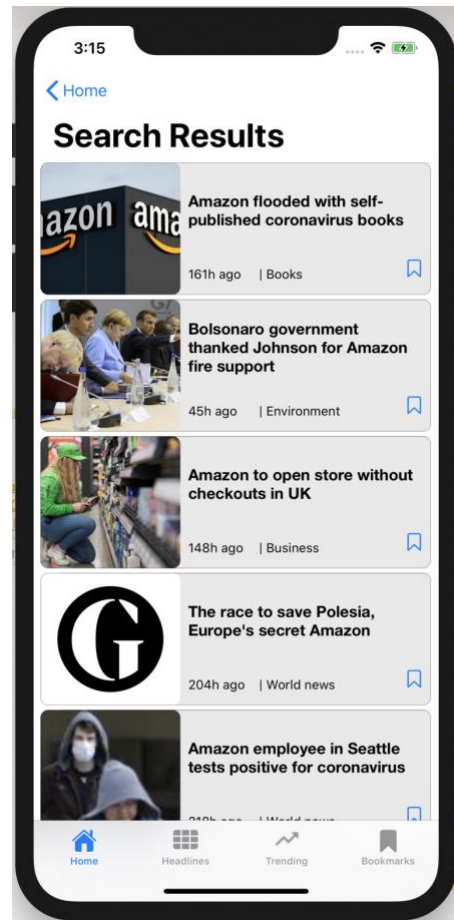


Figure 13: Search Result Page

5.2 Headlines Tab

The Headline tab has two main views:

1. SearchController which works exactly like the Search Controller on the Home Tab
2. XLPagerTabStrip Button Bar Type

5.1.1 XLPagerTabStrip

XLPagerTabStrip is Container View Controller that allows us to switch easily among a collection of view controllers. Pan gesture can be used to move on to next or previous view controller.

The categories to be shown are: **WORLD, BUSINESS, POLITICS, SPORTS, TECHNOLOGY** and **SCIENCE**.

Check out the video to see the exact behavior and the spinners to be added while loading each category.

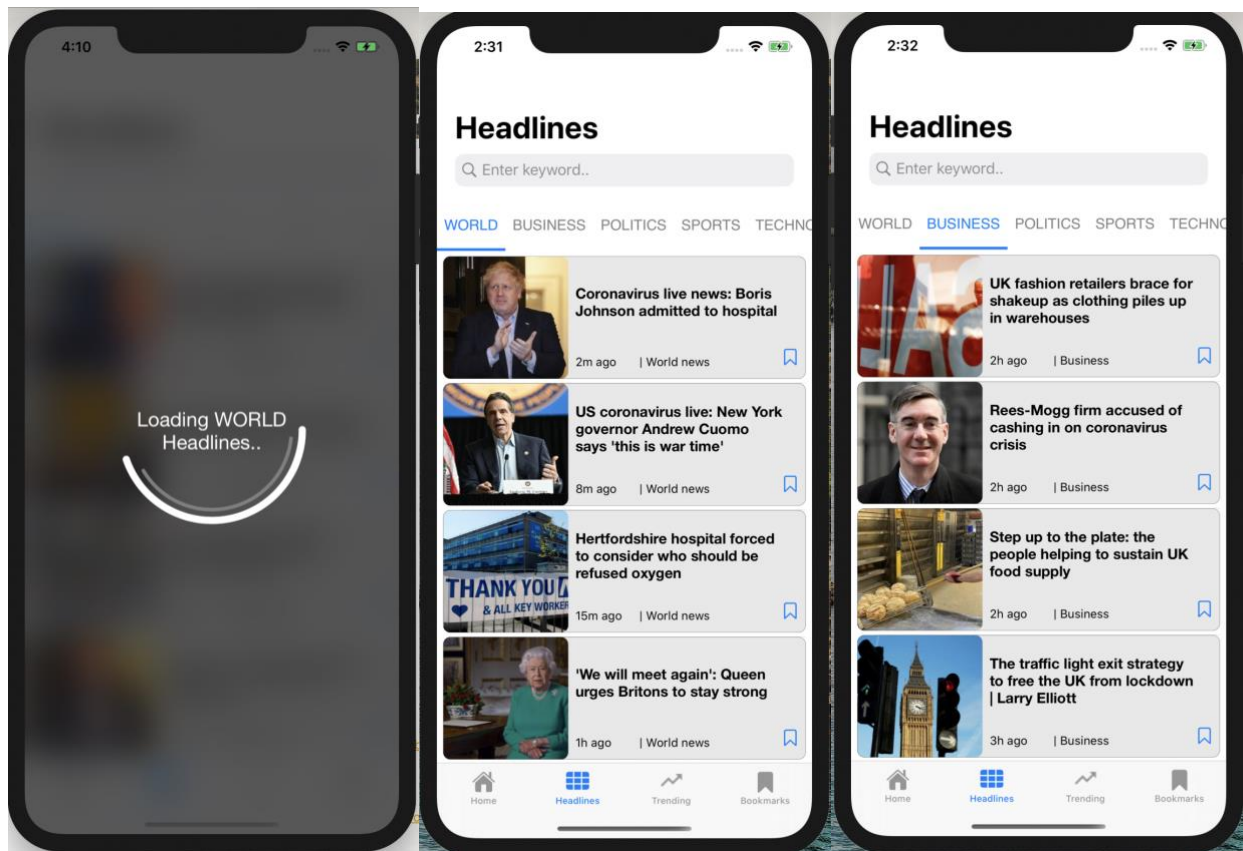
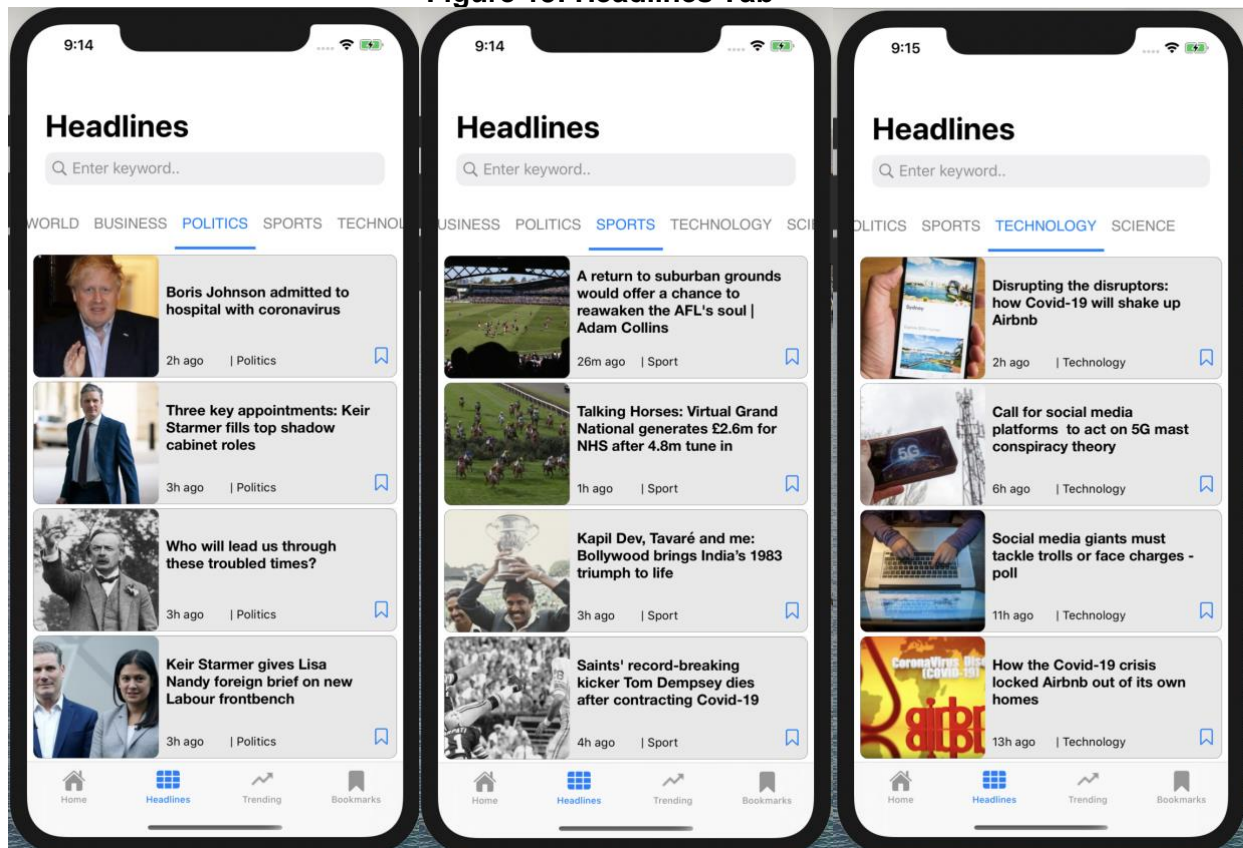
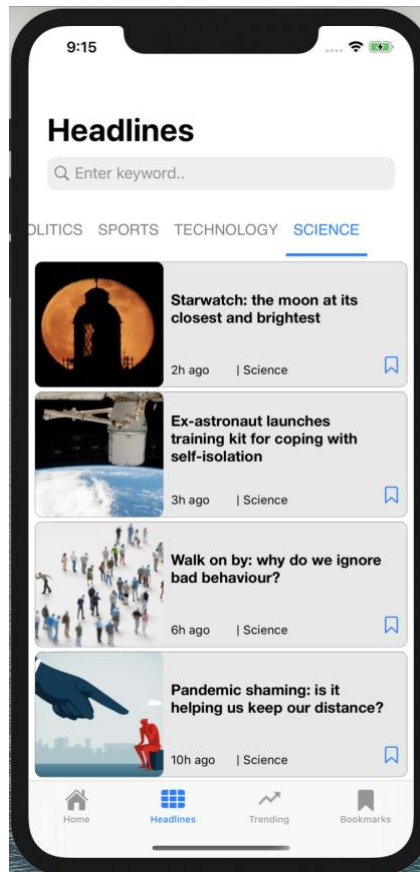


Figure 13: Headlines Tab





The API response for the below API Endpoints: **(same as HW8)**

- **World:** This shows the latest news related to the World News.
The API call to get data for this fragment:
[http://content.guardianapis.com/world?api-key=\[YOUR_API_KEY\]&show-blocks=all](http://content.guardianapis.com/world?api-key=[YOUR_API_KEY]&show-blocks=all)
- **Business:** This shows the latest news related to the Business News.
The API call to get data for this fragment:
[http://content.guardianapis.com/business?apikey=\[YOUR_API_KEY\]&show-blocks=all](http://content.guardianapis.com/business?apikey=[YOUR_API_KEY]&show-blocks=all)
- **Politics:** This fragment shows the latest news related to the Political News.
The API call to get data for this fragment:
[http://content.guardianapis.com/politics?api-key=\[YOUR_API_KEY\]&show-blocks=all](http://content.guardianapis.com/politics?api-key=[YOUR_API_KEY]&show-blocks=all)
- **Sports:** This fragment shows the latest news related to the Sport News.
The API call to get data for this fragment:
[http://content.guardianapis.com/sport?api-key=\[YOUR_API_KEY\]&show-blocks=all](http://content.guardianapis.com/sport?api-key=[YOUR_API_KEY]&show-blocks=all)
- **Technology:** This fragment shows the latest news related to the Technology News.
The API call to get data for this fragment:
[http://content.guardianapis.com/technology?apikey=\[YOUR_API_KEY\]&show-blocks=all](http://content.guardianapis.com/technology?apikey=[YOUR_API_KEY]&show-blocks=all)
- **Science:** This fragment shows the latest news related to the Science News.
The API call to get data for this fragment:
[http://content.guardianapis.com/science?api-key=\[YOUR_API_KEY\]&show-blocks=all](http://content.guardianapis.com/science?api-key=[YOUR_API_KEY]&show-blocks=all)

The properties in the news card in these tabs are as follows:

- **Image:** “file” property of the “assets” array in the “elements” array in the “main” object in the “blocks” object in the “results” array of the “response” json.
- **Title:** “webTitle” property in the “results” array of the “response” json.
- **Time:** “webPublicationDate” property in the “results” array of the “response” json.
- **Section:** “sectionName” property in the “results” array of the “response” json.
- **Bookmark Icon:** On clicking the bookmark icon, that particular news will be bookmarked, and the bookmark icon will change. Similarly, if the news is already bookmarked, it will be removed from bookmarks and the bookmark icon will change back.
- **Article ID:** “id” property in the “results” array of the “response” json. This will be used for the API call to fetch details of each article (Refer section 5.6)

Notes:

- All the News Cells on this page have the same properties as the Home Tab TableView.
- If you still have to change Node backend, make sure you don’t break the React app (if grading is still in progress) or deploy a newer instance of Node.
- If any image is not available, use the default Guardians image provided.

5.3 Trending Tab

The trending tab shows a LineChart implemented using [Charts](#) for interest overtime in a particular search term. The get the data for the chart use the [Google Trends API](#). By default the chart should show the data for the term “Coronavirus”. After entering a different search term and returning (pressing enter) the data should change to the interest over time for that search term as shown in Figures 14 and 15.

Details:

1. Please specify the startTime to be - 2019-06-01, when calling the Google trends API.
2. Extract all the “value” property from the “timelineData” array in the “default” object from the response (Refer the link provided above for a sample JSON response structure).
3. The “value” array extracted is to be set to the LineChart.
4. The Entry is a pair of (x,y) coordinate where x is the counter and y is the value.

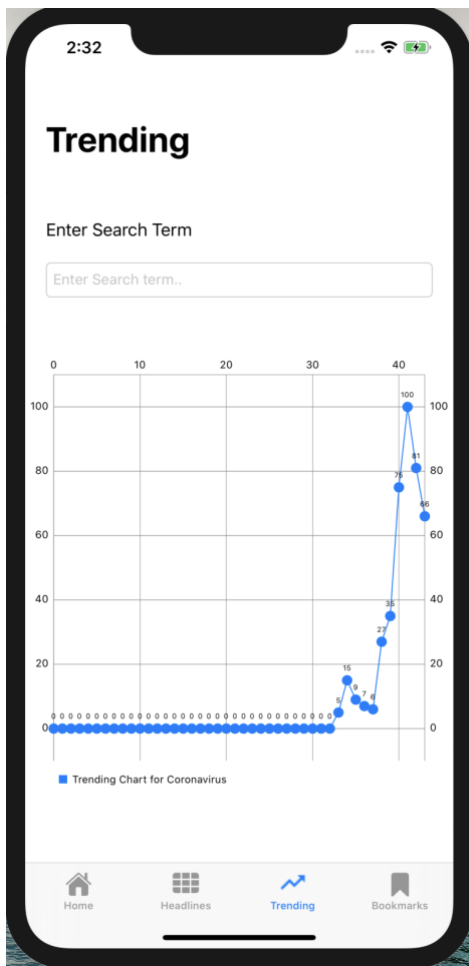


Figure 14: Trending Tab Default

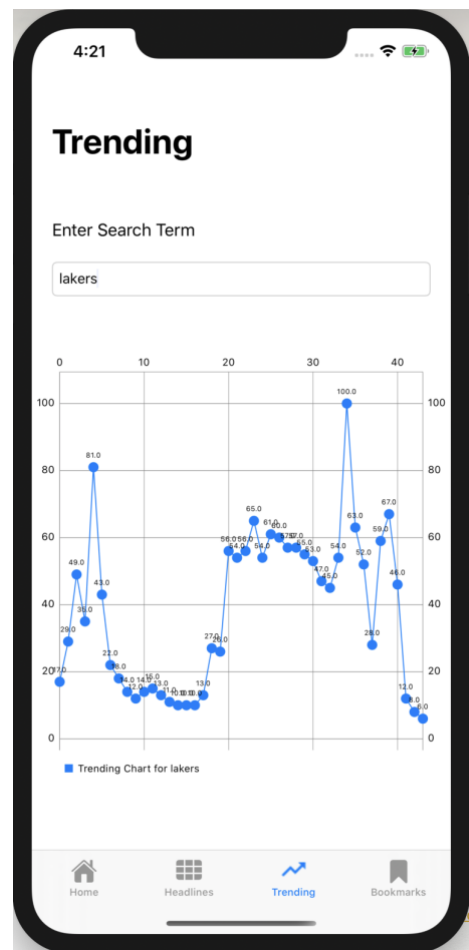


Figure 15: Trending Tab after search

5.4 Bookmarks Tab

The bookmark tab uses UICollectionView to show the Bookmarked news articles as shown in the Figure.

The user should be able to add and remove articles from bookmark from the TableViewCell bookmark button, the Context Menu of Home, Headlines and Search Results Page as well as the Detailed Article Page. They should also be able to remove the article from the Bookmark tab itself from the UICollectionViewCellCell bookmark button. A toast message as shown in figure should be displayed whenever a bookmark is added or removed. Please watch the video carefully. The implementation details for bookmarks are given in the next section.

5.4.1 Adding to Bookmark

The user can add a news article to bookmark from the Home Tab, Headlines Tab, Detailed Article Page and also the Search Results Page. The news can also be bookmarked from the Context Menu.

5.4.2 Removing from Bookmark

The user can remove a news article from the Home Tab, Headlines Tab, Detailed Article Page, Search Results Page and also from the Bookmark tab. The news can be also removed from bookmark from the Context Menu.

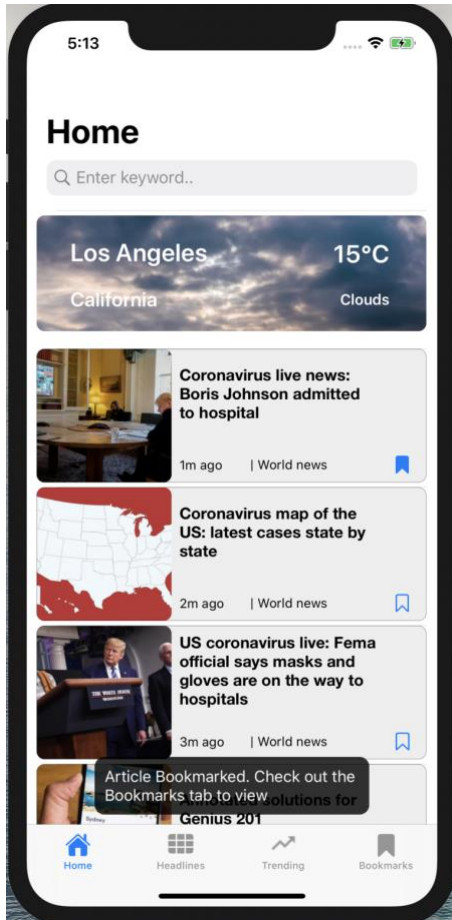


Figure 16: Added to bookmark

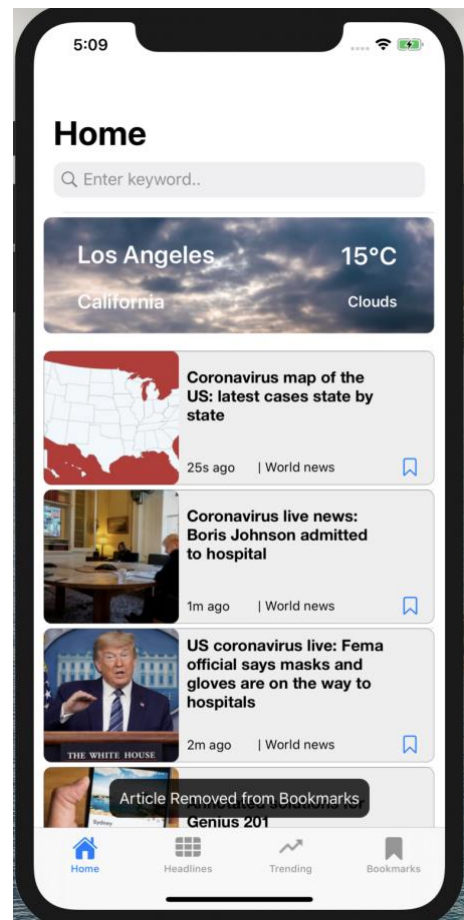


Figure 17: Removed from bookmark

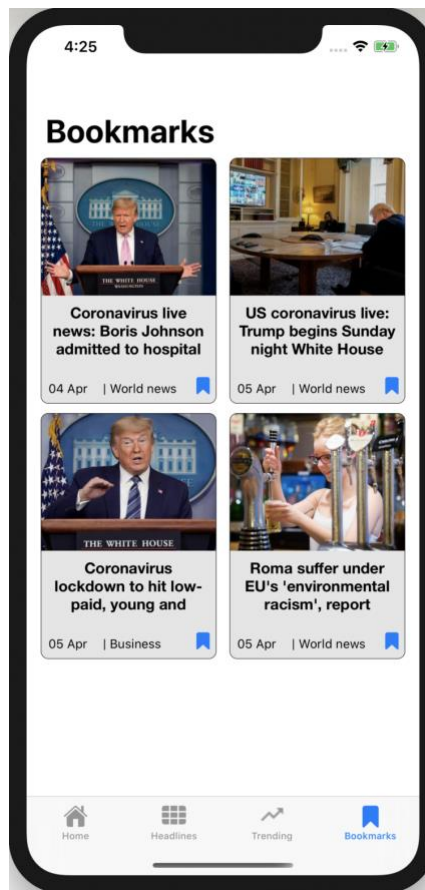


Figure 18: Bookmarks Tab

5.4.3 Empty Bookmark Tab

When no article is bookmarked show the message “No bookmarks added” as shown in Figure.

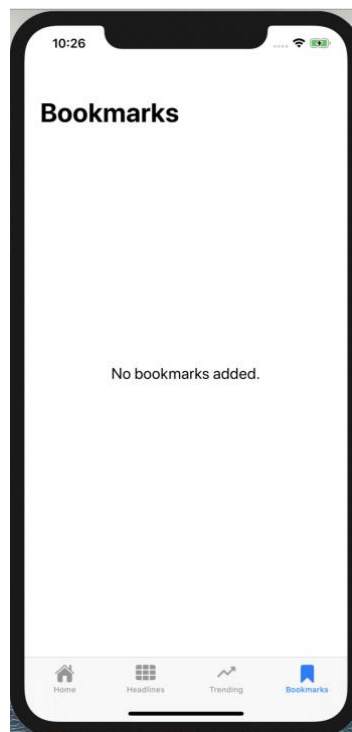


Figure 19: Empty Bookmarks Tab

5.4.4 Bookmarks Implementation

Use “**UserDefaults**” to store bookmarked articles data.

<https://developer.apple.com/documentation/foundation/userdefaults>

Key Points to note :

1. The bookmarked list should persist even after closing the app. The items should be removed from bookmarked list only if the user removes them (see video).
2. You may need to pass data between different views. You can refer this link to learn about data passing between Views in Swift : <https://learnappmaking.com/pass-data-between-view-controllers-swift-how-to/> .
3. The horizontal scroll of the views should replicate exactly as shown in the video

6. Twitter Share

Share with Twitter (Twitter icon) to share the article URL on Twitter. Once the button is tapped, a web page should be opened (in Safari) to allow the user to share the Article information on Twitter, as shown in Figure 20. Note the article URL and the #CSCI_571_NewsApp in the twitter message.

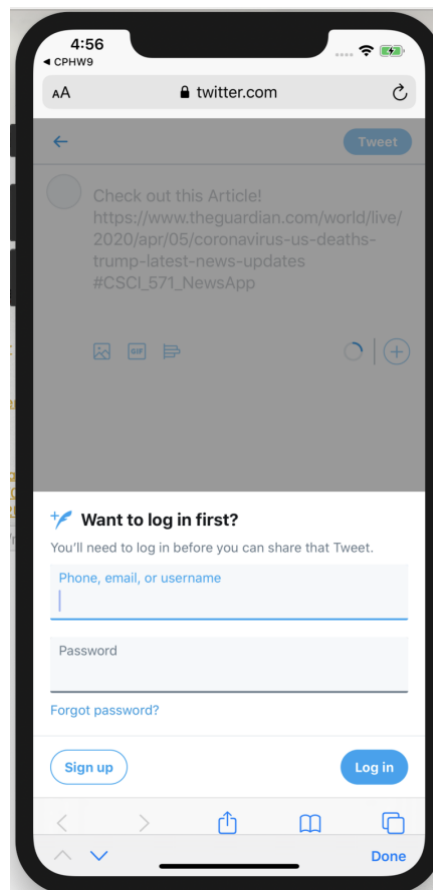


Figure 20: Twitter Share Screen

7. Additional Information

For things not specified in the document, grading guideline, piazza, or the video, you can make your own decisions. But keep the following points in mind:

- Always display a proper message and don't crash the app, if an error occurs.
- You can only make HTTP requests to your backend (Node.js script on AWS/GAE/Azure) except for the Bing Autosuggest API, which can be called directly from your Swift code.
- All HTTP requests should be asynchronous.

8. Hints

- Pull Down Refresh: <https://cocoacasts.com/how-to-add-pull-to-refresh-to-a-table-view-or-collection-view>
- Context Menus for iOS 13: <https://kylebashour.com/posts/ios-13-context-menus>
- Scroll View for Detailed Page: <https://www.youtube.com/watch?v=KmE50giVuLA>

9. Images / Icons

[Click](#) to download the images needed for this HW. **All** the icons needed for the app are included in the downloads. Add the Assets directly to your app to access all the images.

10. Material You Need to Submit

Unlike other exercises, you will have to demo your submission using Zoom Remote Control during a special grading session. Details and logistics for the demo will be provided in class, on the Announcement page and on Piazza.

Demo is done on a MacBook using the emulator, and not on a physical mobile/tablet device.

You should also ZIP all your source code (without image files and third-part modules) and push the resulting ZIP file by the end of the demo day to GitHub Classroom.

****IMPORTANT**:**

All videos are part of the homework description. All discussions and explanations in Piazza related to this homework are part of the homework description and will be accounted into grading. So please review all Piazza threads before finishing the assignment.