Assignment 3
Mobile Analytics Platform

In this part of the assignment, you will create a mobile analytics platform to collect information about how customers are using your application. The platform should collect and aggregate the following information:

- **User Session.** A user session is defined as the amount of time an application is continuously active. For example, if a user opens the app uses it for 30 seconds and then switches to another app, this would be a single session of length 30 seconds. Later that day, if the user opens the app and uses it for 180 seconds, that would be a second user session of 180 seconds.

- **User Events.** A user event is recorded when the user performs a specific action in the application. For example, you may choose to record button taps, segue to specific view controllers, or the numbers of times that a user reloaded a page. *For this application and simplicity, we will only be concerned with a button tap user event.*

- **Touch Heat Map.** Our app is designed for single handed use and our UI/UX team is considering how the screens size of the "plus" phones is effecting the usability of our app. We want to collect all the touches that a user makes in a given session.

Every day at 10pm, you should send an email to the developer with a summary of the apps usage statistics. The information in the email should include the following:

- **Daily Active Users.** This represents the number of unique users that used your app on a given day.

- **Daily New Users.** The number of new users for a given day.

- **Monthly Active Users.** The number of users for the current month.

- **Daily Sessions.** The number of sessions for a given day.

- **Average Session Length.** The average length of session for a given day

Flow of information

The following lays out the flow of information between the mobile device and the backed.

On the Mobile Device

User Action	Data to Record	Where to Store	Send to Backend
User launches the application	Session start timestamp	User defaults	No
User taps a button	Button identifier, timestamp	User defaults	No
User navigates to another view controller	View controller name, timestamp	User defaults	No
User closes the app	Session end timestamp	User defaults	Yes

On the device, you should store the information locally in UserDefaults until the session is over. When the user closes the app, you should post the data to your analytics platform. There are many different ways you could post the data, but a good strategy would be to cover the data to JSON and post it as the HTTP body. The following shows an example of the data collected:

```
let data = [
  "session" : [
    "user"  : "some_unique_identifier",
    "start" : "Apr 11, 2017, 6:17 PM",
    "end"   : "Apr 13, 2017, 6:19 PM",
    "events" : [
      [ "id" : "share-button" ],
      [ "id" : "home-button"  ],
      [ "id" : "info-button" ],
      [ "id" : "navigate-home" ],
      [ "id" : "navigate-home" ]
    ],
    "touches": [
      ["x": 10.5,"y" : 0],
      ["x": 20.1,"y" : 20],
      ["x": 30.2,"y" : 11],
      ["x": 110.6,"y" : 311],
      ["x": 0.2,"y" : 1],
      ["x": 310.9,"y" : 2]
    ]
  ]
]
```

The following provides shows how to convert the data to JSON and post to a local development server (http://localhost:8080).

```
// Create the url with URL
let url = URL(string: "http://localhost:8080/")!

// Create the session object
let session = URLSession.shared

// Create the URLRequest object using the url object
var request = URLRequest(url: url)

// Set http method as POST and add headers
request.httpMethod = "POST"
request.addValue("application/json", forHTTPHeaderField: "Content-Type")
request.addValue("application/json", forHTTPHeaderField: "Accept")

// Convert the data to JSON
do {
  let jsonData = try JSONSerialization.data(withJSONObject: data,
options: .prettyPrinted)
  request.httpBody = jsonData
} catch {
  print(error.localizedDescription)
}

//create dataTask using the session object to send data to the server
let task = session.dataTask(with: request as URLRequest,
                            completionHandler: { data, response, error in
                              // This is what should be returned by the server
                              print(response)
                              print(data)
})
task.resume()
```

In this example the data is passed to the server as JSON through the request body.

On the Backend

Once the data is posted to the backend, it should be added to the datastore. Set up a cron job to execute a task that queries the data store and sends out an email each night at 10:00 PM CST.   Make sure that you have set up the proper indexes to query the data (this is done automatically when you run the query in the development environment). The email can be sent from the default "appspotmail.com" domain.

The following is an example of the email:

```
Dear Developer,
Here is your daily summary for your application:
 *Daily active uses: 11
 *Daily new users: 1
 *Monthly active users: 65
 *Daily sessions: 36
 *Average session length: 70 seconds

Have a great evening!
```

While you will be only sending a single session data at time during development, you will need to design the backend to handle the million (hopefully) of user sessions happening concurrently. You should use best practices as discussed in class.

Requirements

The focus of this assignment is the backend design and development. You may consider this document to be guidelines to follow. You may choose an alternate strategy for any part as of the assignment as long as you are able to send the nightly email with correct information.

There are no specific requirements for the mobile application used to test the analytics platform. You may use a previous app, an Xcode Playground, or a command line script to simulate user sessions. We will be building an iOS app for the next assignment, so you may want to refresh your memory.

Please include the application, Playground or scripts with your assignment so that we can test it accordingly. If you choose to use a Playground or script, you should ensure that the data is randomized so that the same values are not being submitted.

The assignment is Due Thursday, October 17th at 5:29 PM.

Bonus Point and Enduring Admiration

A bonus point and my enduring admiration will be awarded to applications that generate a daily heat map graph and include it in the email. I would recommend looking at the Google Charts API which can be used to generate an image. The image can be included as an attachment in the mail.

The heat map should honor the dimensions of a device screen (iPhone 7 pictured below) and highlight the most frequently tapped areas.