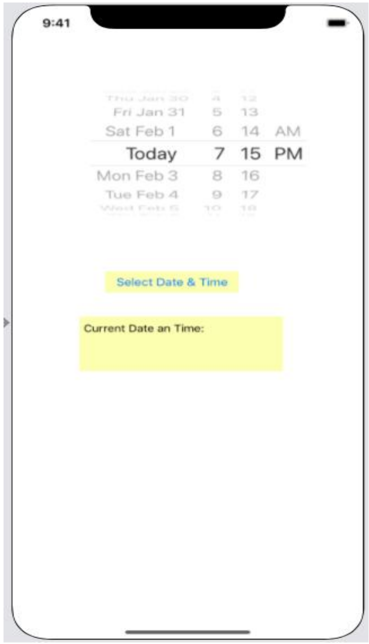
In-class Activity:

1. Create a new single view iOS App with XCode. Add a DatePicker object, a Button and a TextView into the storyboard's GUI panel like this below:



2. Connect all three GUI controls to the ViewController.swift file:
   - Both the DatePicker and TextView objects are connected as new referencing outlets, using a name like datePicker and dateDisplayView respectively.
   - The button should send an event like "touch up inside" to a method (give it a name like getDateTime).

3. Initialize the DatePicker with the current date and time as soon as the app is loaded with a view. Add the following statements inside the ViewDidLoad() method:

```
let date = Date()
datePicker.setDate(date, animated: false)
dateDisplayView.text = "Current Date and Time: \(date)"
```

4. Complete the method getDateTime() with the following code:

```
let date = datePicker.date
dateDisplayView.text = "Date & Time selected: \(date)"
```

5. Run the app! The date output doesn't look good, eh!

6. Format the date output to be displayed in the TextView. Replace the last statement inside getDateTime() method with the following code:

```
 let dateFormatter = DateFormatter()
 dateFormatter.dateStyle = .medium
 dateFormatter.timeStyle = .short

 dateFormatter.locale = Locale(identifier: "en_US")
 dateDisplayView.text = "Date & Time selected:\n" +
dateFormatter.string(from: date)
```

7. Run the app again to see the difference!

8. We can also extract individual component from a date. Try to add the following statements at the bottom of the getDateTime() method:

```
let components = datePicker.calendar.dateComponents([.day, .month, .year,
.hour, .minute], from: datePicker.date)
let day = components.day!
let month = components.month!
let year = components.year!
let hour = components.hour!
let minute = components.minute!
dateDisplayView.text += "\nDate created from Components:
\(year)/\(month)/\(day) -  \(hour):\(minute)"
```

9. Run the app again! Take a screen shot of your simulator and submit it.