IOS Programming Fundamentals – MAD 3115

**AVANT** **** ..

**A MAD3115 FINAL PROJECT**

**Student Name**  : *Rosette Lopez Pardillo*

**Student ID**  **:** *C0768425*

**Course :** *MAD3115*

**Schedule :** *12:45 126BM*

Overview

MADT 3115 Final project requires to submit a functional, deliverable Single View App IOS Application

**Avant**. Is a simple IOS app that schedules user emails and notify user to send it.

Purpose

In this busy technological era, people tend to forget a lot of things even a simple greeting to ones dear to them.

**Avant** will remind users to send emails as scheduled in just a tap!

Functionality and Features

The message created from the Compose Message View will be displayed in the Message List. Once the message schedule is due, it will send notification with actions *Ignore* and *Send It!*. Choosing *Send it!* will action to send the email and transfer message details from Message List view to History view. If user opt for Ignore, no action will be processed.

**Message List**

The first view a user will see once **Avant** is opened is the messages list

This is implemented using Table View Controller with prototype cells.

This page has 3 buttons

* *ellipsis.circle* – allows to reorder the arrangement of messages displayed on the page
* *square.and.pencil* – allows the user to create a new messages. This will transfer to another view
* *go.backward* – allows the user to view the list of sent messages. This will transfer to another view

Parts of a message cell

* *Message Title* – Displayed as a cell title. This is the subject of the message that is created
* *Recipient* – Displayed as cell detail as a subtitle. The email of the person to whom the messages is for
* *Detail Accessory* – Tapping this will pop the time remaining before message is set to be sent.

Other actions:

Tapping a specific message cell will bring the user to modify the message details in another view.

**Compose Message**

This is implemented using a Table View Controller with static cells and 2 sections.

* First section contains:
  + To – *TextField*. the recipient email address
  + Message Title – *TextField* . The subject of the message or email
  + Schedule – *TextField* . The schedule when the user wants to send the message. When cell is tapped, DatePicker will be displayed adding another cell. If tapped again, DatePicker is hidden
* Second section contains
  + Message Body – *TextField*. It is where the message content will be inputted

An alert pops as a confirmation from the user to set the message. There are two actions available for this alert. Cancel does nothing while Set it! sets the message.

Another alert pops if text fields are left empty when the user Set it!’s the message.

The default schedule of the date picker is the current date and time

**History Message List**

This is implemented using Table View Controller with prototype cells and displays a list of all sent messages.

Tapping a specific message cell will bring user to another view which will display the message details

**Sent Message Details**

This is implemented using UI View Controller and will just display message details in a label

**Technologies**

Notifications. Due messages appear to users as notifications. The notification is implemented using Framework UserNotification

Email. Class MFMailComposeViewController is used for the app to be able to send email to recipeints.

How to email with ***AVANT***

Requirements:

* Iphone
* Internet (for email sending)

Additional setup required:

* User email account must be added in Iphone
  + To add email, go to Settings->Passwords & Accounts->Add Account

1. Open **Avant** App. Make sure to allow **Avant** to send notifications. Message list is empty on first use
2. Tap square.and.pencil icon positioned on the top left part of the view to compose new message. This will display the compose message view
3. Fill in all fields such as recipient email, message title, schedule and message. Once done, tap paper plane icon to set message.
4. An alert will pop out as a confirmation on message creation. Tap cancel to cancel setting or tap Set it! to set the message.
5. Go back to the Message list and the newly composed message is listed
6. Tapping the Message list will enable user to modify the message details
7. Tapping the accessory icon will let the user know the time left when the message is due
8. Once a message is due for sending, it will send notifications to remind user to send it. Notification runs on foreground and will pop even when **Avant** is not opened.
9. Notifications gives user two action options. (a) Ignore – which will ignore message and the message can still be viewed, modified and deleted from the Message list view, and (b) Send It! which will bring user to Apple Mail View and user presses send to send email
10. If user opts for other options (ie. Cancel, save draft) when in Apple Mail view, the user can still see the message in the Message List view and can send, modify or delete message
11. If notifications were ignored, user can send due email by tapping the accessory icon from the Message list view
12. If the email is sent, the message details can be viewed from the History view by pressing the backward icon populated at the bottom right part of the Message list view
13. When in History view, user can see a list of sent message and by tapping each message, user can see the message details

Difficulties Encountered

1. As a developer, I have been implementing code following requirement specification. Now that I have to do things on my own, I wasted time on scraping what I have developed half a day because I consistently change my mind
2. Email Sending – This app was supposed to send email automatically without user interaction but it is against Apple Guidelines. As another way, a notification will pop on user device to remind user to send the email with just a tap.
3. Email Sending – Just as the xcode project was functioning as expected, emails were not sent when configured in the college. It turns out using the Cestar wifi blocks the email message sending
4. Date Picker – Took time to implement a date picker and needed the aid of other github user implementation to make it work
5. TapGesture – DidSelectRow was not that responsive when adding a tapgesture on the views, specifically in Compose View where a cell needs to be tapped to show DatePicker. It took time to find out that the solution was just a single line setting cancelTouchesInView to false.

Github

Link: <https://github.com/otetLopez/avant>

Repository: avant

Branch: master

References

* DatePicker populated in a cell implementation:

<https://github.com/rajtharan-g/InlineDatePicker>

* Email Sending:

<https://www.hackingwithswift.com/example-code/uikit/how-to-send-an-email>

* Setting Notifications:

<https://cocoacasts.com/actionable-notifications-with-the-user-notifications-framework>

Files Submitted

Main ZIP file ***A\_MAD3115\_GP.zip*** that contains the following:

* Report[PDF] ***Rosette\_C0768425\_MAD3115\_FP.PDF***
* Presentation[PPTX] ***Avant Presentation 2.PPTX***
* Xcode Project ***Avant***  folder