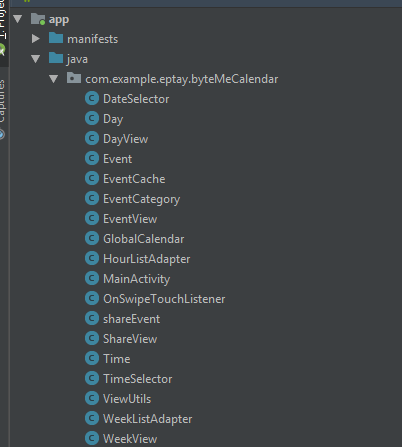
BYTE ME CALENDAR DOCUMENTATION

1. Within our project, we utilize the “Singleton” class design pattern. We implemented two classes that uses the singleton pattern: EventCache and GlobalCalendar.

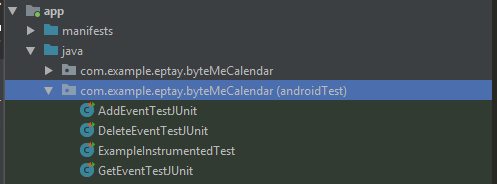
We chose to use the singleton pattern on these two classes because MonthView, WeekView, DayView, and EventView needed to access these classes to generate data. We also used the singleton pattern because the other classes needed to store and remove events to the calendar. It would have been inefficient for each class to have its own instance of these two classes. It also would have been difficult to index stored events if each view had its own cache of events. Having only one instance of a class that is accessed globally, removes the need for each view to send and receive information. Making the calendar global removed the need for unnecessary object creation.

The EventCache class can be found in the file extension:

app/java/.com.example. eptay. byteMeCalendar/EventCache. The GlobalCalendar class can be found in the file extension: app/java/.com.example. eptay. byteMeCalendar/GlobalCalendar



1. The four test cases can be found in the file extension: app/java/.com.example. eptay. byteMeCalendar(androidTest). Within it there are the classes: AddEventTestJUnit, DeleteEventTestJUnit, GetEventTestJUnit and (ADD LAST ONE HERE).



The AddEventTestJUnit Class test 4 cases:

1. An event with correct information is added and is test to see if it is added correctly.
2. An event is added with null values to see if it will be added to the cache.
3. An event is added with an invalid hour value
4. An event is added with an ending time that is before the starting time

The DeleteEventTestJUnit Class tests 4 cases:

1. An event is added with correct information and is removed.
2. Attempts to delete an event that does not exist.
3. A repeating event that repeats daily is added and removed
4. A repeating event that repeats monthly is added and removed

The GetEventTestJUnit Class tests 4 cases:

1. Attempts to retrieve a valid event
2. Attempts to retrieve an event that does not exits
3. Attempts to retrieve an event that was removed
4. Attempts to retrieve an event that has an invalid time frame

NEED 4th TEST CLASS

1. Building and running the program:

* Ran on Nexus 6, One Plus 6t, Google Pixel. UI might be obstructed on other lower resolution devices
* Clean build in Android studio
* Run app to start ByteMe Calendar
* Click floating action button to add event
* Hold event title in Month view to open up menu for edit, delete, share
* Share functionality only works if app is ran on an actual android device. Also please run share functionality twice because the first time, android studio will ask to allow sms messaging.
* Click menu button at the top left corner to access navigation drawer
* Click Week view to go to the week of the selected day in the Month view
* In Week view, click any day to go to the day view of the selected day
* Click Day view to go to the day view of the selected day in Month view
* In Month view, swipe right or left to access next or previous month
* In Week view, swipe right or left to access next or previous week
* In Day view, swipe right or left to access next or previous day

Implemented Functionalities:

* Monthly View
  + Ability to add, edit, share and delete events
  + Highlights current day
  + Swipe left or right to change months
  + Select a date and open the day view for selected date
  + Open navigation drawer
* Navigation Drawer
  + Move to Month view.
  + Move to Weekly view.
  + Move to Daily view.
* Adding and Editing Event View
  + Setting start and ending time
  + Setting a title and description
* Weekly View
  + Swiping left and right to change weeks
  + Selecting a day on the weekly view to view in daily view
* Daily View
  + Displaying Events based on their starting and ending time
* Sharing Events
  + Sending event information to a phone number