name: Jun Li
uid: 1418447
partner: Chun-Hao Hsu

1. Each person pick a piece of code from your own project (class, method, etc)

We both merge the anroid view with composables. The composbales made code less and clean, however mixed frameworks made layouts a liitle hard to control.

Although the transition from android view to jetpack compose ease our lives but in this project we find that drawing on Canvas and saving the bitmap locally is easier to handle on custom view compared to composable. So, we both retain the custom view in phase1.

In our composables, we should less refer to some gloable variables, which would make testing harder. So we will make our functions clearner and “light-weighted” by adding more callbacks or mutable states.

2. Pick a piece of functionality that both of your projects include. Discuss the similarities/differences between your implementations.

We implement the custom view with similar ideas but one used drawPath method and one used drawShape method. Draw path will draw more smoothly like really pens. Both are good practices.

3. Pick part of your code that you don't think is well tested.

We both lack the model part tests(like the database and repository). Using Room class inMemoryDatabaseBuilder can help to build database in memory to test our DB.

```
db = Room.inMemoryDatabaseBuilder(
    ApplicationProvider.getApplicationContext(),
    DrawingDatabase::class.java
).build()
```

The Mockito libary can help to mock some objects to help testing repository.

```
// Mockito to mock  dao.addDrawData method
testCoroutineScope.runBlockingTest {
    drawingRepo.addDrawingToDB(filename, filepath)

    // veryfiy dao.addDrawData is called
    Mockito.verify(mockDrawDao).addDrawData(
        DrawData(timestamp = any()
            , filename = filename, filepath = filepath)
    )
}
```

Here we use Mockito to mock dao.addDrawData method and // veryfiy dao.addDrawData is called.