

Phase 2 Code Review

Muyuan Zhang

2023-11-09

discussed with Levi Neely

Improvement

`e.printStackTrace()` was mainly used when we were debugging, but some of the error messages could be helpful for the user as well. Levi's group brought up a great usage of `Toast` which enables the user to see the result of their actions or error messages during using the app.

Similarities/Differences Between Implementations

In Levi's project, all the Kotlin classes are under the same package.

In `ClickFragmentContent` (the gallery composition), a `LazyColumn()` was created and the thumbnails of all the existing paintings are listed in it.

We have a different structure for all the Kotlin classes: we split the classes into several packages (`gesture`, `data`, `model`, `components`, etc.) according to their contents, which simplified debugging and cooperation.

In order to mimic the UI of Instagram, we utilized the `LazyVerticalGrid()` to arrange the squared thumbnails and keep the images in the order of update time by saving the last save timestamp and storing them into a self-defined composable `DrawingListItem()`.

Testing

We should have included the tests for the database to test creating a database, closing the database as well as creating/deleting/updating drawings:

```
@Before
    fun createDb() {}
@After
    fun closeDb() {
        db.close()
    }
@Test
    fun testCreateDrawing() {}
@Test
    fun testDeleteDrawing() {}
@Test
    fun testUpdateDrawing() {}
```