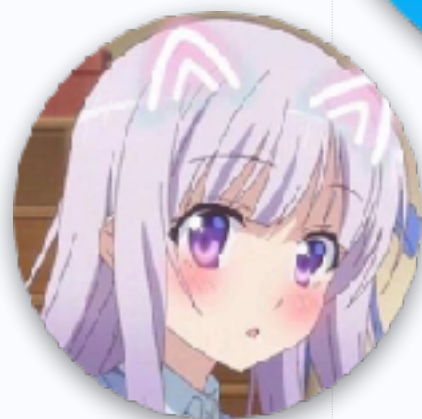




龚玥
11611908



关浩
11612831



钟兆玮
11611722



张淮化清
11510598

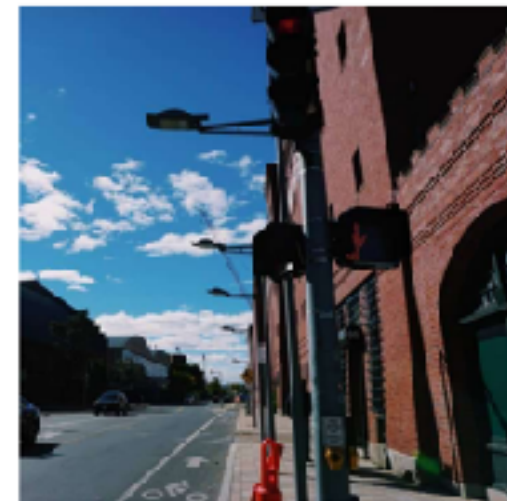


周颖
11610701

INTRODUCTION

Pilipili is a photo and image processing app, which can apply artistic styles to photos or videos in real-time. here are at least 11 different styles from Van Gogh's Starry Night to Picasso's Les Femmes d'Alger. Also, users can upload and share their own creative work to community.

PiliPili



TIMELINE

Week 10	Upload a photo/video, Save photo to local photo album
Week 11	Take a picture using a camera
Week 12	Photo style transfer, Provide various filters for photos...
Week 13	Share photos in community, Thumb up a photo or video in the community...
Week 14	Test, Upload to Andriod app store...
Week 15	Review issues, Fix bugs, Add new features...

VERSION HISTORIES

 [sustech-se2019](#) / [main-project-repository-pilipili](#) Private

 Watch ▾

4

★ Star

0

 Fork

0

 Code

 Issues 0

 Pull requests 0

 Projects 0

 Wiki

 Security

 Insights

Releases

Tags

[Draft a new release](#)

11 days ago 

version-2.0 

update readme

 02f9084  zip  tar.gz

on Apr 20 

version-1.0 

update readme


 54f99a9  zip  tar.gz



FEATURES

1. User register/login

PiliPili



Username


root

Password

LOGIN

No account yet? Create one

PiliPili



Name

zh

Password


Re-enter Password

Password Do not match

CREATE ACCOUNT

All PiliPili: Signup Failed login

PiliPili



Name

中文名字


Password

Re-enter Password

CREATE ACCOUNT

All PiliPili: Unknown Error login

PiliPili



Name

♪♪♪♪

Password

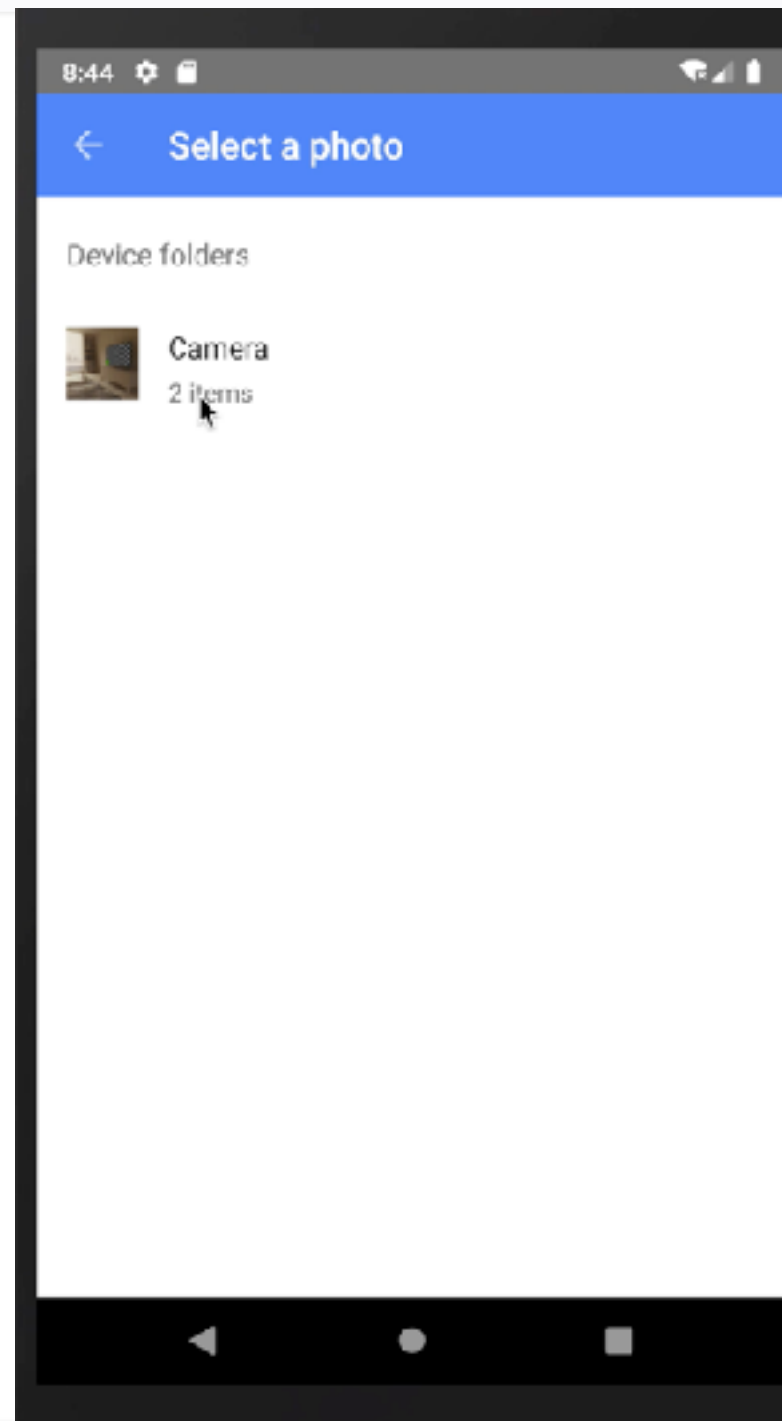
Re-enter Password

CREATE ACCOUNT

All PiliPili: Unknown Error login

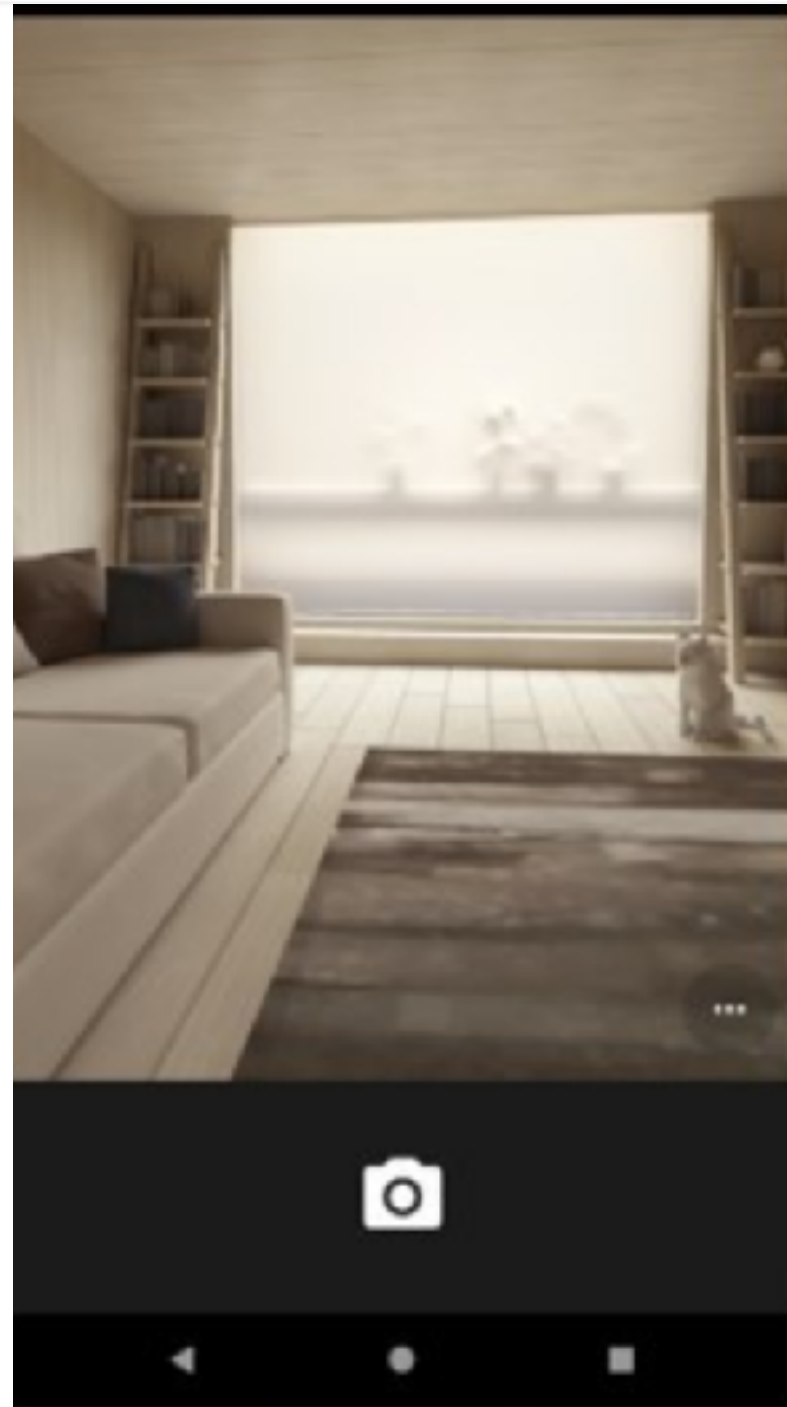
FEATURES

2. Upload a photo or video



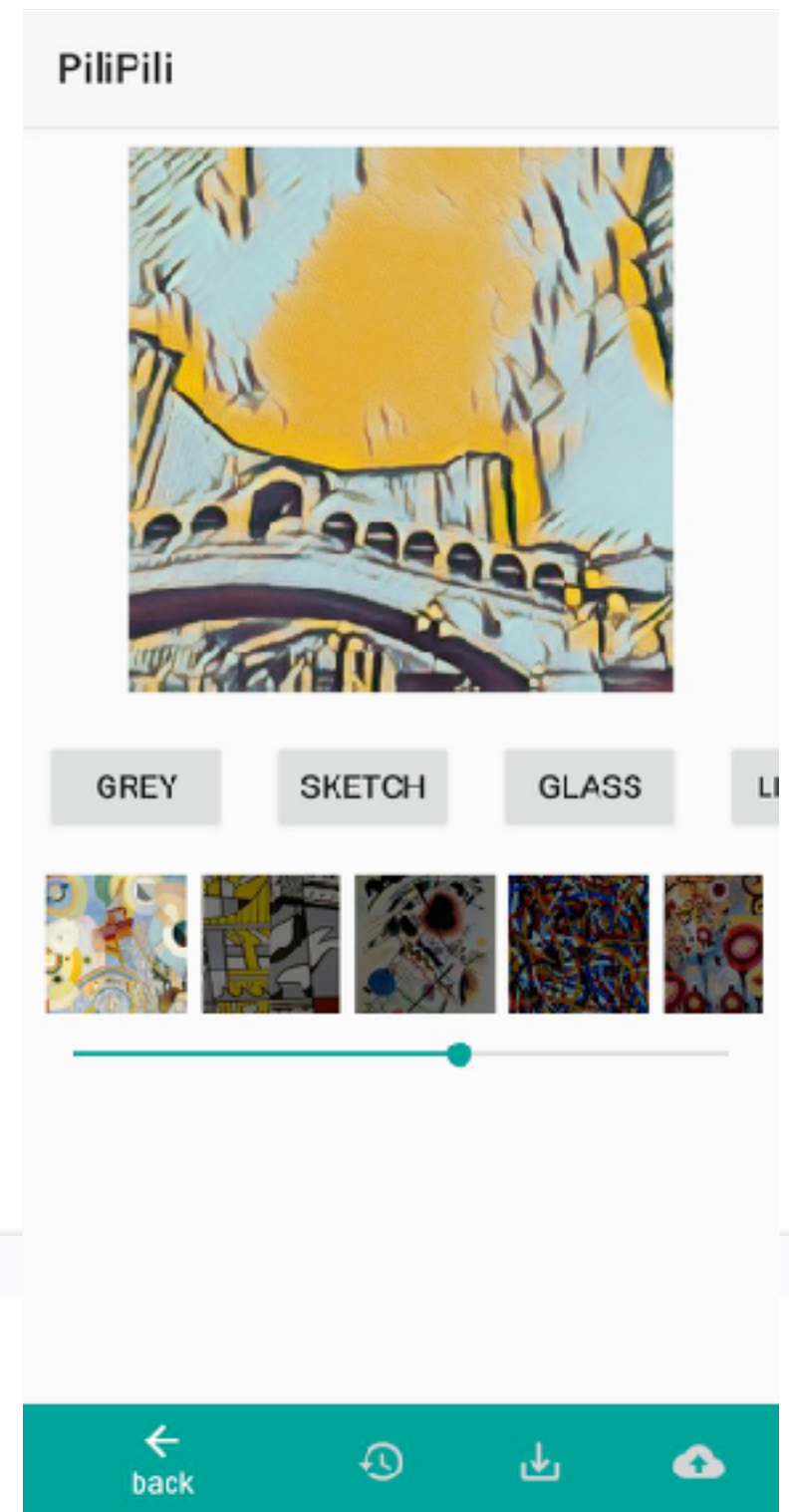
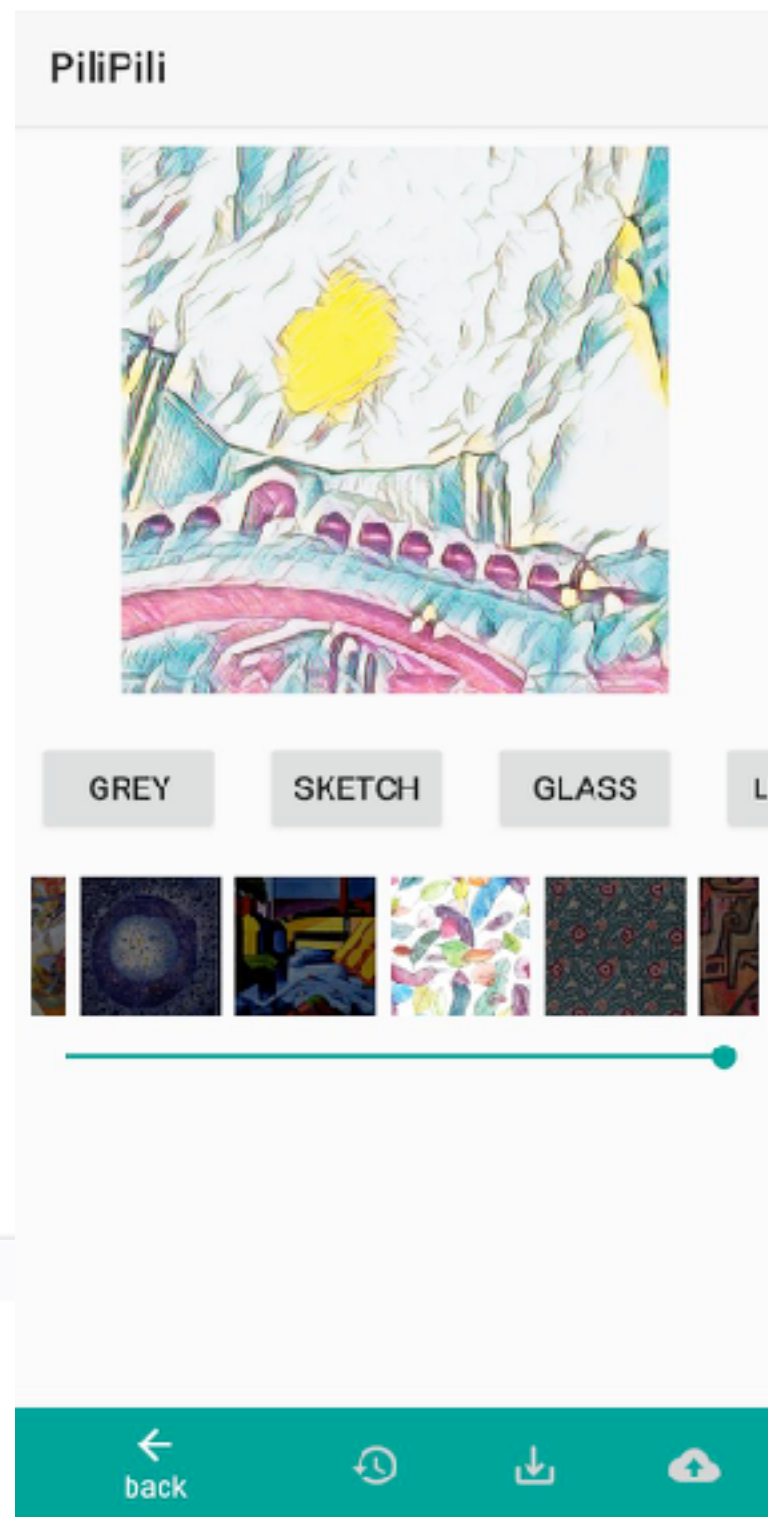
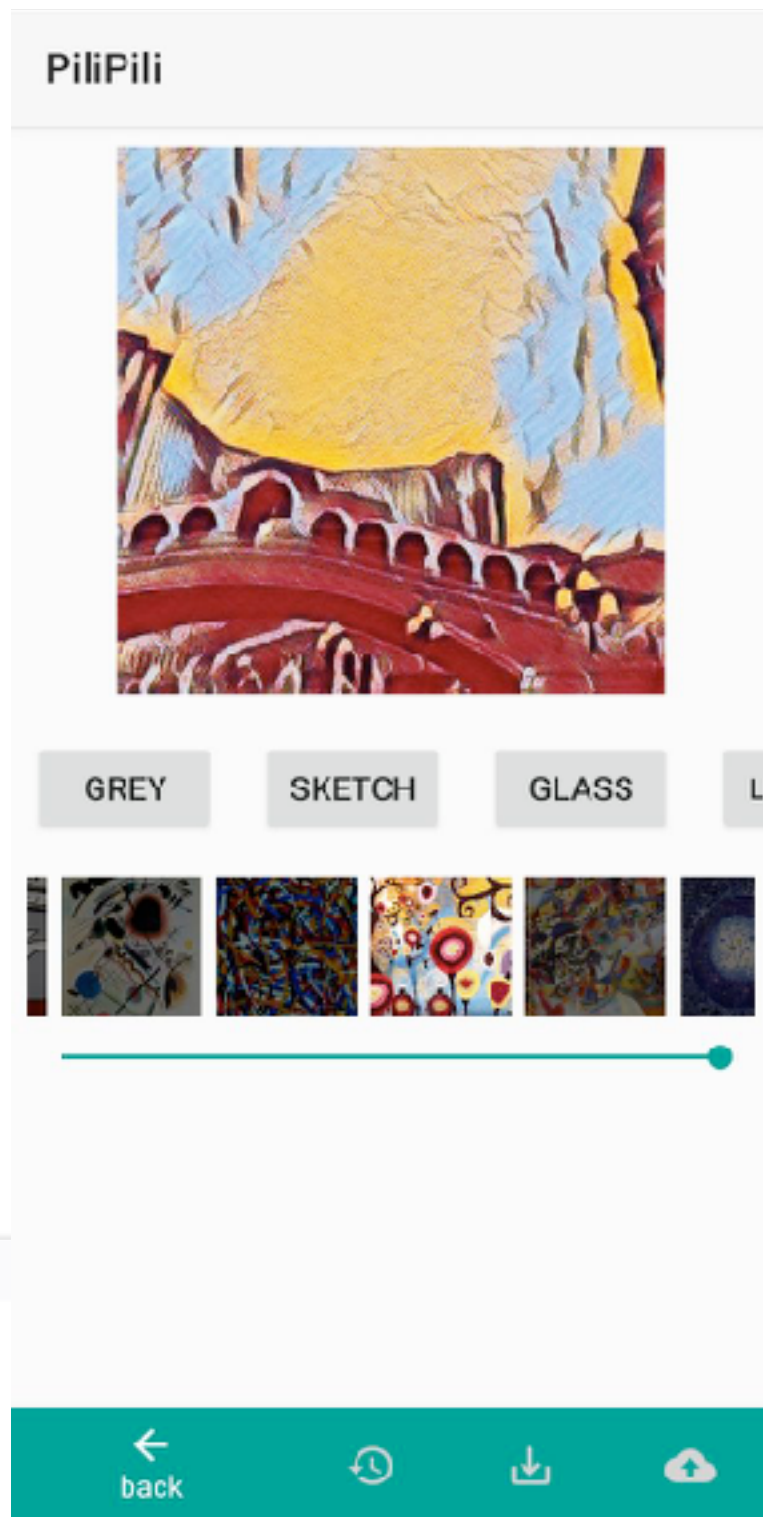
FEATURES

3. Take picture using a camera



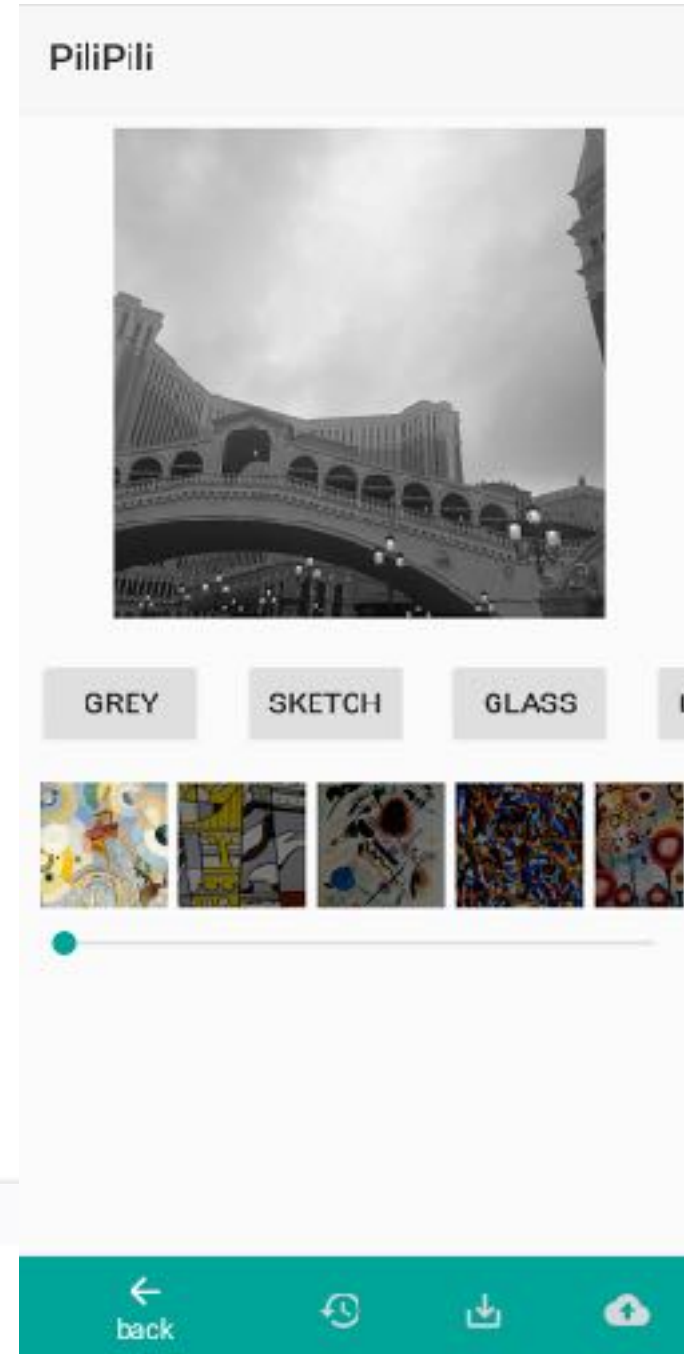
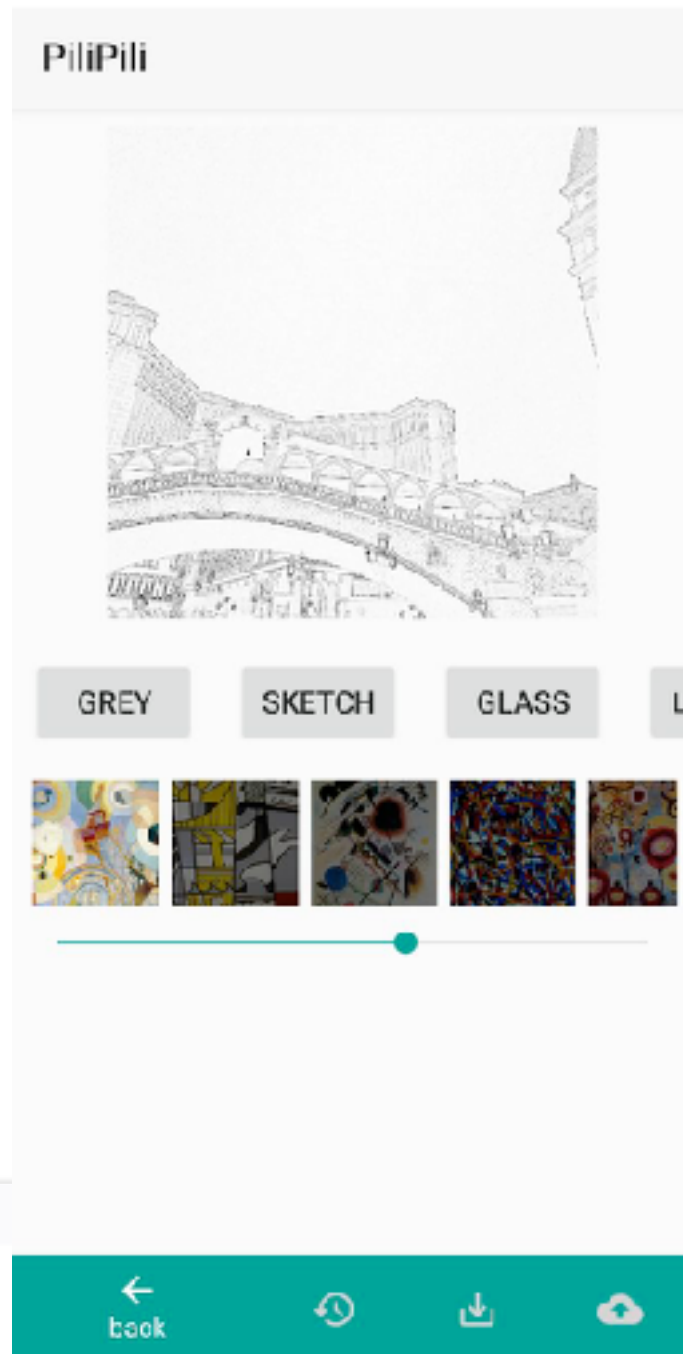
FEATURES

4. Photo style transfer



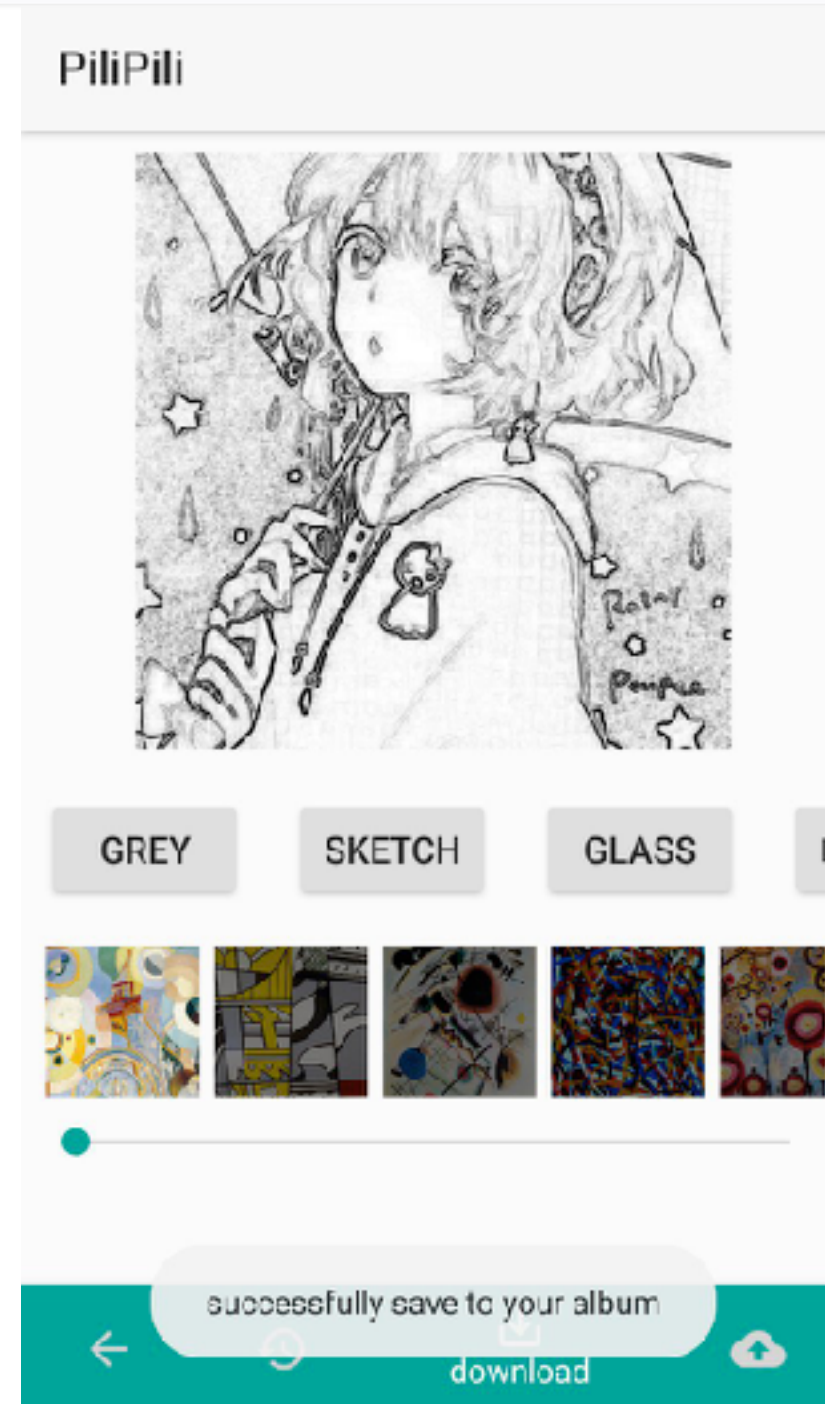
FEATURES

5. Provide various filters for photos



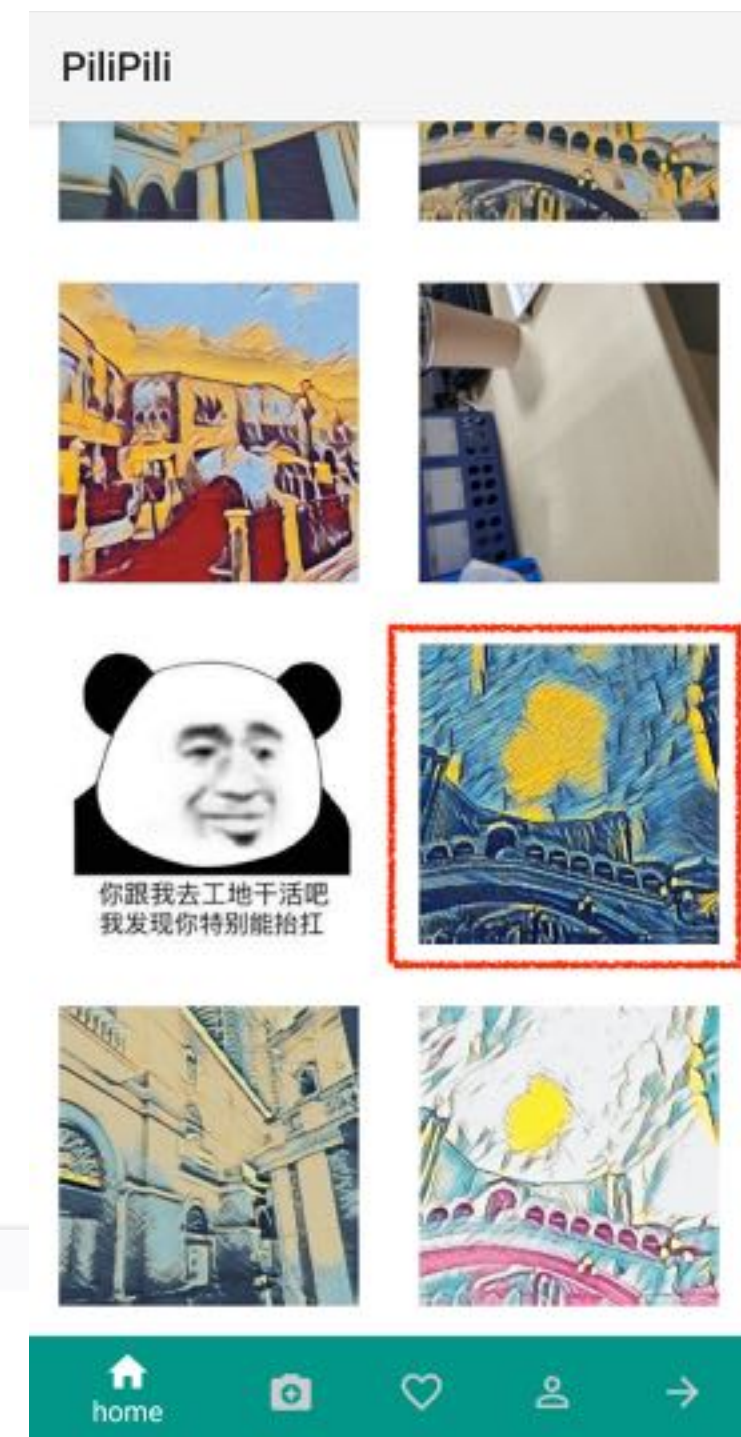
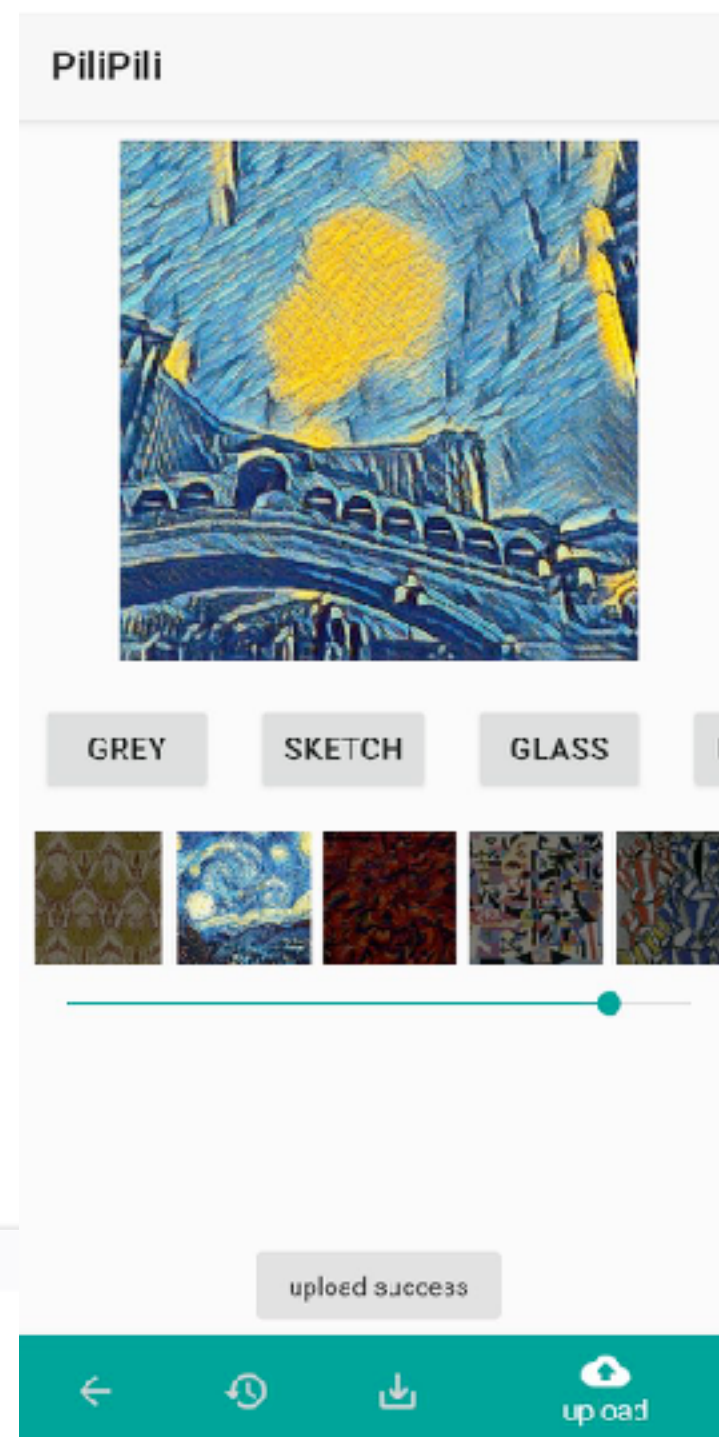
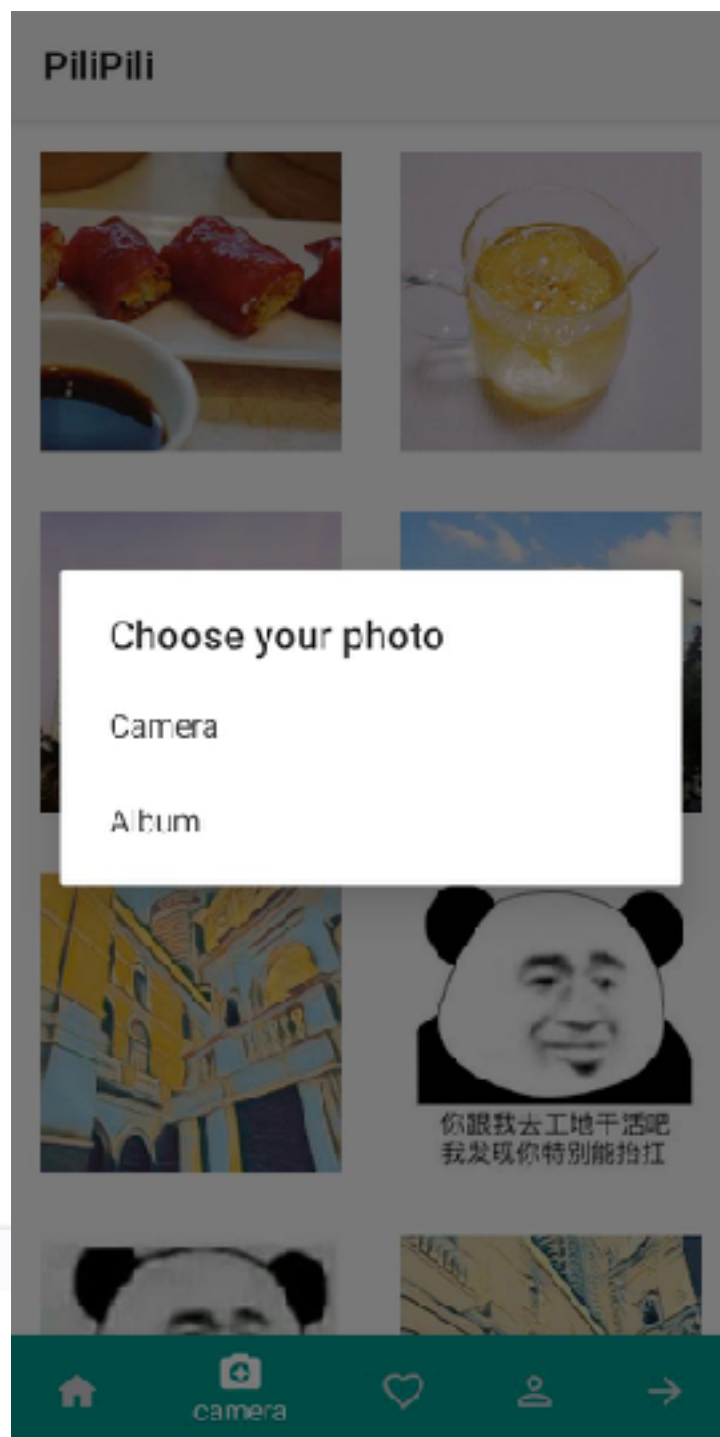
FEATURES

6. Save photo to local photo album



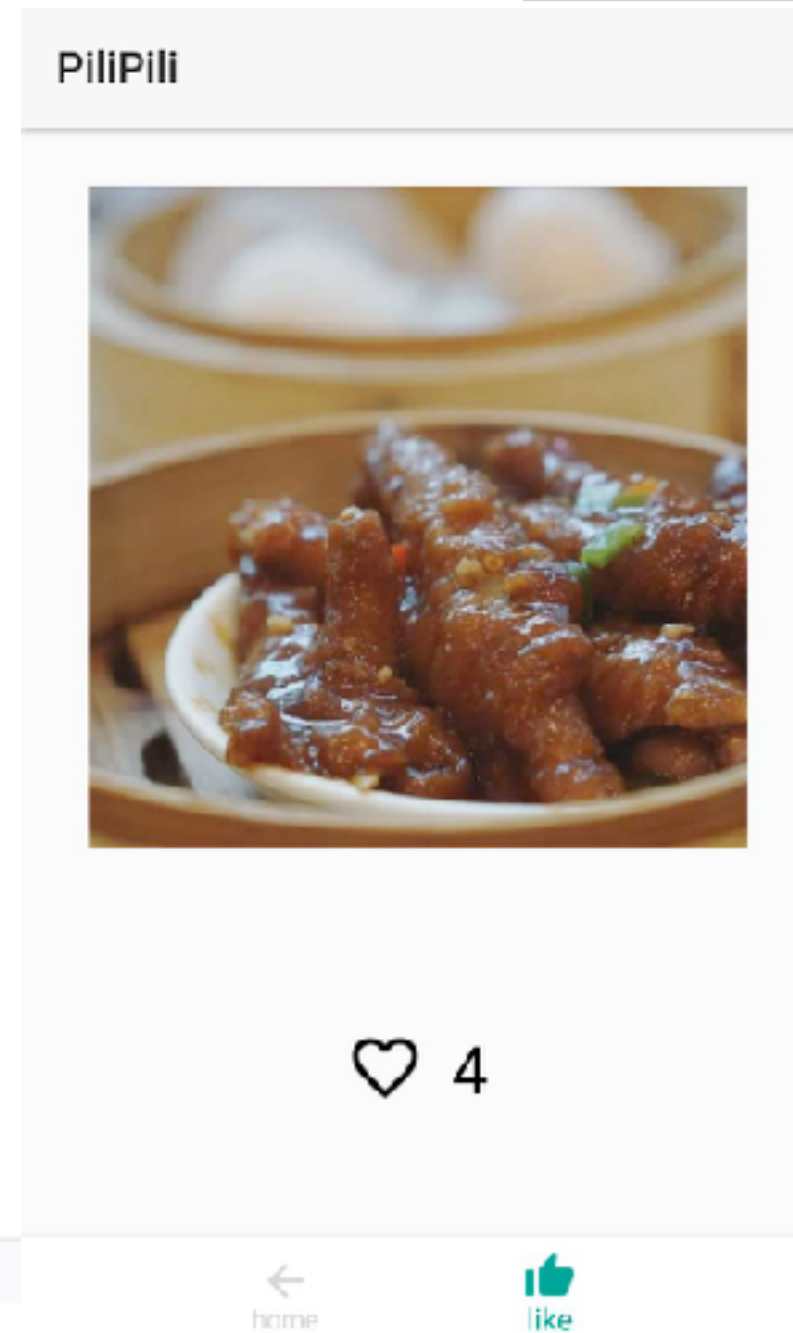
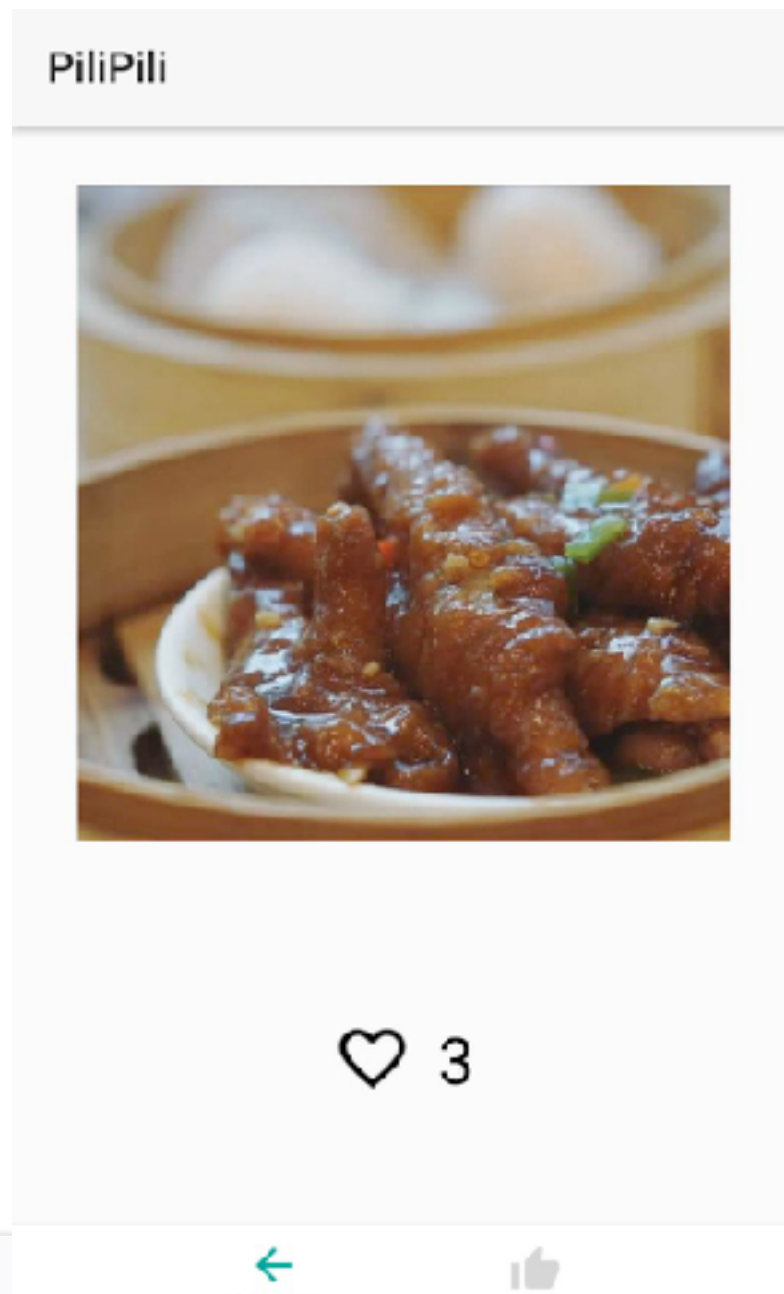
FEATURES

7. Share photos in community

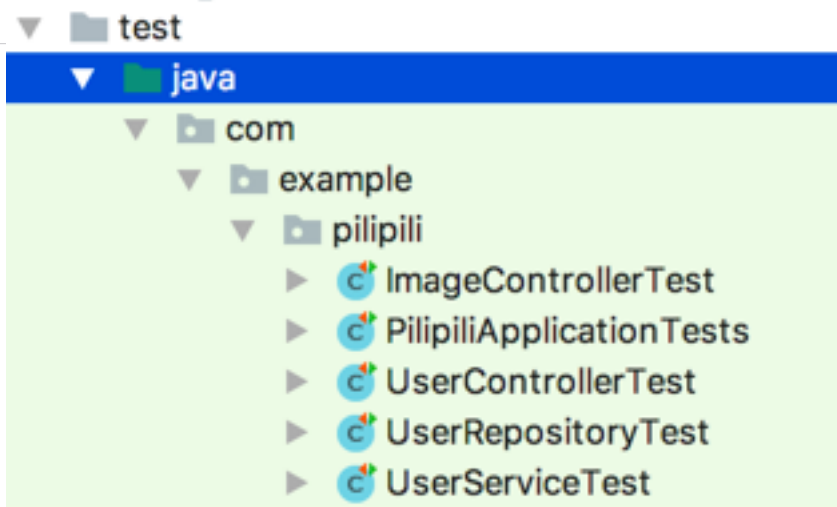


FEATURES

8. Thumb up a photo or video in the community



TEST: JUNIT + MOCKMVC



```
public class UserControllerTest {
    @Autowired
    private MockMvc mvc;

    @Autowired
    private WebApplicationContext context;

    @Before
    public void setupMockMvc() throws Exception {
        mvc = MockMvcBuilders.webAppContextSetup(context).build();
    }

    @Test
    public void testLoginSuccess() throws Exception {
        String str = "{\"userName\":\"root\",\"password\":\"1234\"}";
        System.err.println(str);
        byte[] outputBytes = str.getBytes( charsetName: "UTF-8");
        mvc.perform(post( urlTemplate: "/userLogin")
            .contentType(MediaType.APPLICATION_JSON)
            .content(str)
            .accept(MediaType.APPLICATION_JSON))
            .andExpect(status().isOk())
            .andExpect(jsonPath( expression: "$.code").value( expectedValue: 0));
    }
}
```

94% classes, 60% lines covered in package 'com.example.pilipili'

Element	Class, %	Method, %	Line, %
api	100% (0/0)	100% (0/0)	100% (0/0)
config	100% (2/2)	100% (2/2)	100% (9/9)
model	100% (7/7)	73% (34/46)	70% (59/84)
service	100% (5/5)	75% (18/24)	53% (56/104)
web	66% (2/3)	75% (9/12)	56% (47/83)

TEST: ESPRESSO

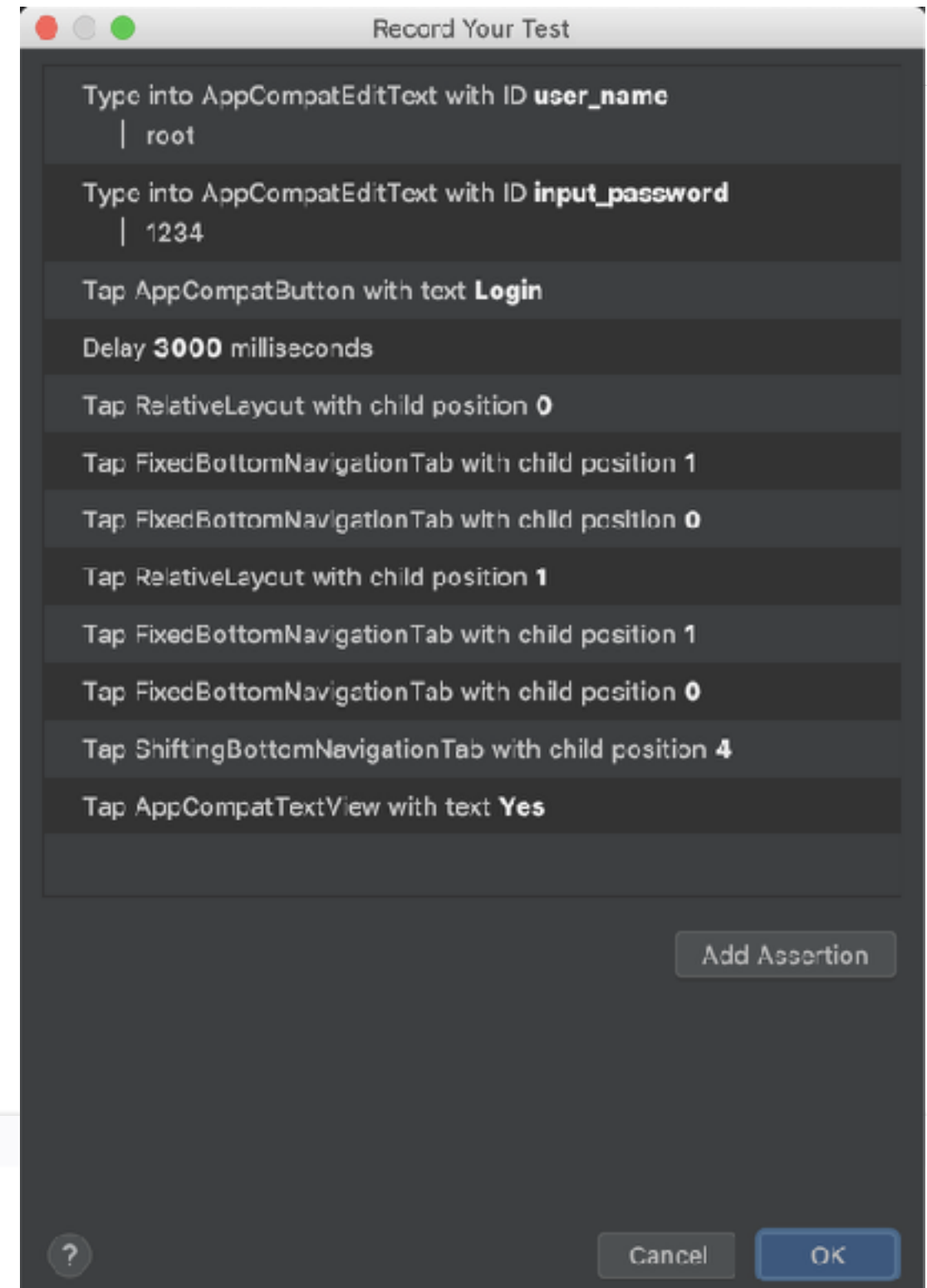
```
> Task :app:transformNativeLibsWithMergeJniLibsForDebugAndroidTest
> Task :app:processDebugAndroidTestJavaRes NO-SOURCE
> Task :app:transformResourcesWithMergeJavaResForDebugAndroidTest
> Task :app:validateSigningDebugAndroidTest
> Task :app:packageDebugAndroidTest
> Task :app:assembleDebugAndroidTest

> Task :app:connectedDebugAndroidTest
Starting 14 tests on MI 6 - 8.0.0

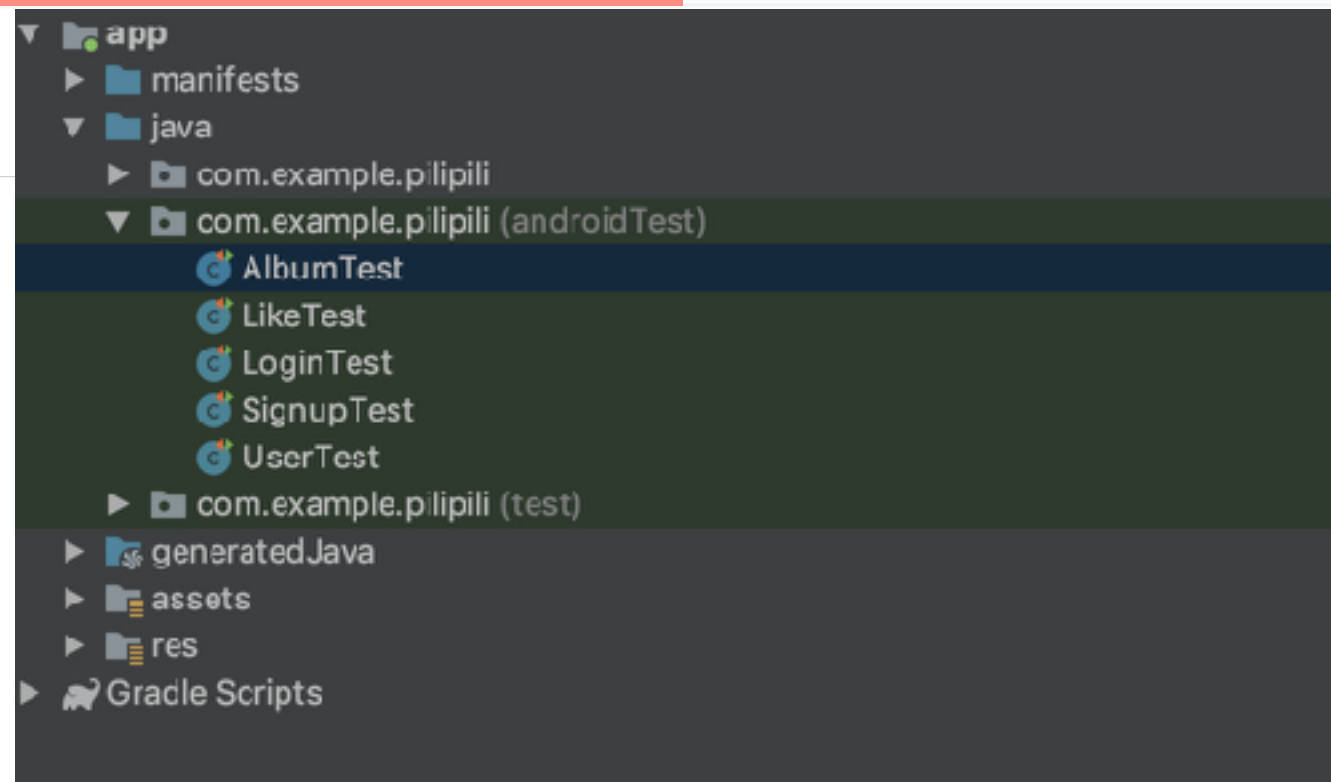
> Task :app:createDebugAndroidTestCoverageReport

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.
Use '--warning-mode all' to show the individual deprecation warnings and suppress them if desired.
See https://docs.gradle.org/4.10.1/userguide/command\_line\_interface.html#sec:command\_line\_warnings for more details.

BUILD SUCCESSFUL in 1m 12s
51 actionable tasks: 10 executed, 41 up-to-date
23:45:00: Task execution finished 'createDebugAndroidTest'
```



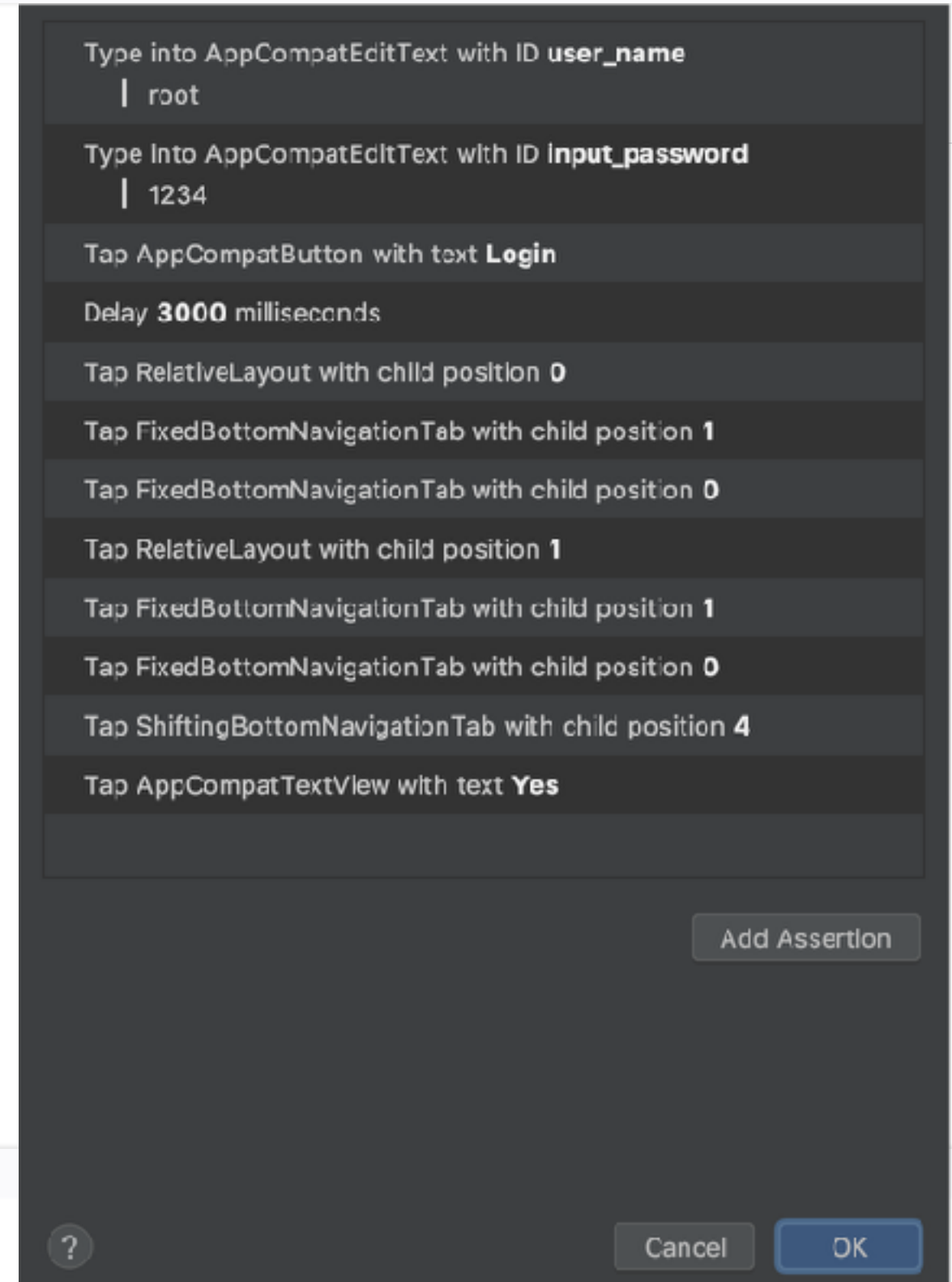
TEST: ESPRESSO



```
@Test
public void albumTest() {
    ViewInteraction appCompatEditText = onView(
        allOf(withId(R.id.user_name),
            childAtPosition(
                childAtPosition(
                    withClassName(isValue("android.support.design.widget.TextInputLayout")),
                    position(0),
                    position(0))
            ),
            position(0))
    );
    appCompatEditText.perform(scrollTo(), replaceText(StringToBeSet("root"), closeSoftKeyboard()));

    ViewInteraction appCompatEditText2 = onView(
        allOf(withId(R.id.input_password),
            childAtPosition(
                childAtPosition(
                    withClassName(isValue("android.support.design.widget.TextInputLayout")),
                    position(0),
                    position(0))
            ),
            position(0))
    );
    appCompatEditText2.perform(scrollTo(), replaceText(StringToBeSet("1234"), closeSoftKeyboard()));

    ViewInteraction appCompatButton = onView(
        allOf(withId(R.id.btn_login), withText("Login"),
            childAtPosition(
                childAtPosition(
                    withClassName(isValue("android.widget.ScrollableView")),
                    position(0),
                    position(0))
            ),
            position(0))
    );
    appCompatButton.perform(scrollTo(), click());
}
```



TEST: ESPRESSO

```
> Task :app:transformNativeLibsWithMergeJniLibsForDebugAndroidTest
> Task :app:processDebugAndroidTestJavaRes NO-SOURCE
> Task :app:transformResourcesWithMergeJavaResForDebugAndroidTest
> Task :app:validateSigningDebugAndroidTest
> Task :app:packageDebugAndroidTest
> Task :app:assembleDebugAndroidTest

> Task :app:connectedDebugAndroidTest
Starting 14 tests on MI 6 - 8.0.0





> Task :app:createDebugAndroidTestCoverageReport

Deprecated Gradle features were used in this build, making it incompatible with Gradle 5.0.
Use '--warning-mode all' to show the individual deprecation warnings and determine if they should be addressed.
See https://docs.gradle.org/4.10.1/userguide/command\_line\_interface.html#sec:command\_line\_warnings for more details.

BUILD SUCCESSFUL in 1m 12s
51 actionable tasks: 10 executed, 41 up-to-date
23:45:00: Task execution finished 'createDebugAndroidTestCoverageReport'
```

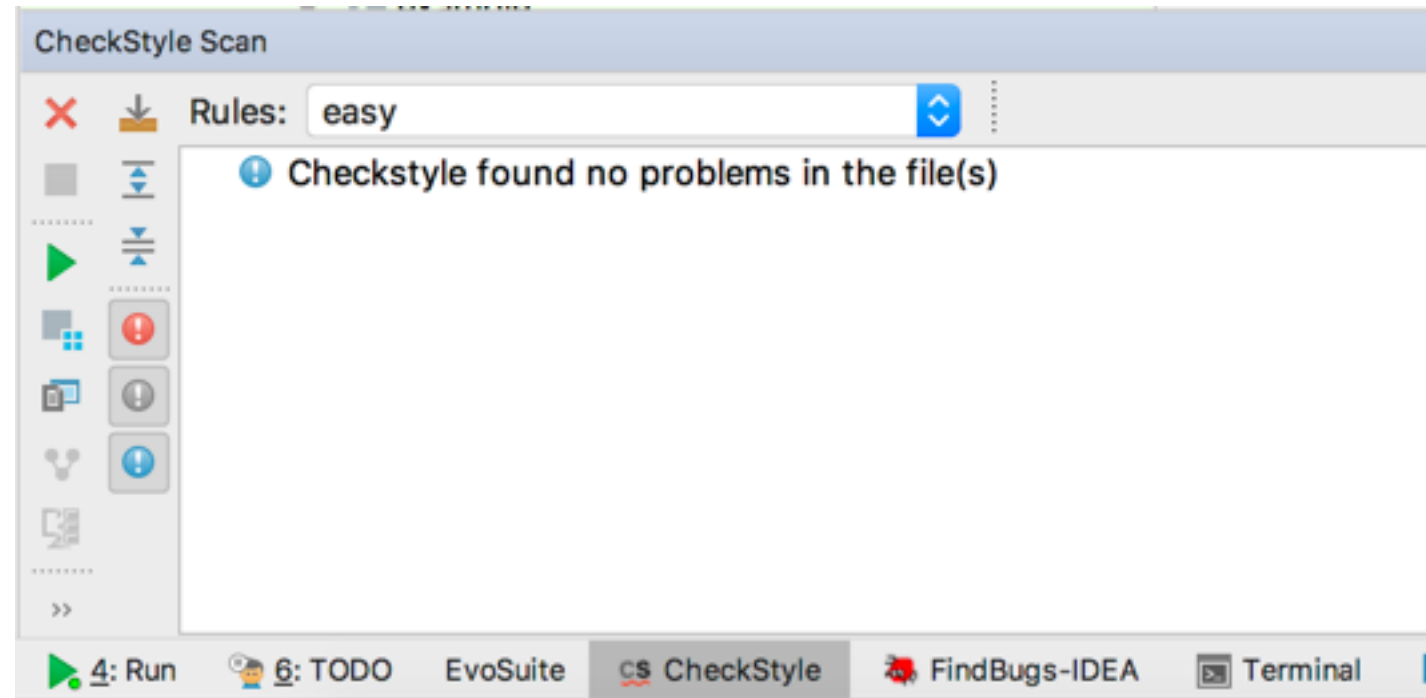
 debugAndroidTest

debugAndroidTest

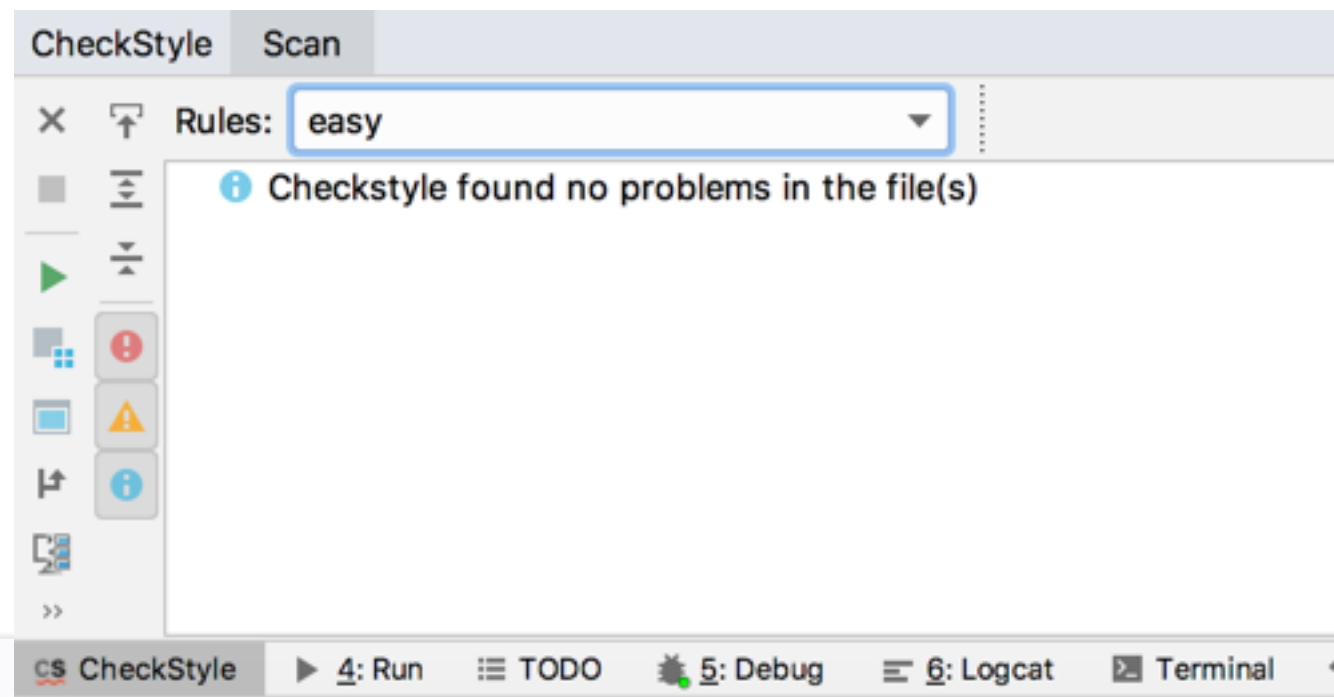
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
 com.example.pilipili	<div><div></div></div>	31%	<div><div></div></div>	26%	173	242	540	822	72	122	18	35
 com.example.pilipili.utils	<div><div></div></div>	44%	<div><div></div></div>	25%	16	24	51	95	11	18	3	5
 com.example.pilipili.service	<div><div></div></div>	81%	<div><div></div></div>	50%	22	53	35	149	14	42	4	15
 com.example.pilipili.model		48%		n/a	4	9	8	13	4	9	0	1
Total	2,940 of 4,934	40%	181 of 253	28%	215	328	634	1,079	101	191	25	56

STATIC ANALYSIS RESULTS: CheckStyle

Backend



Frontend

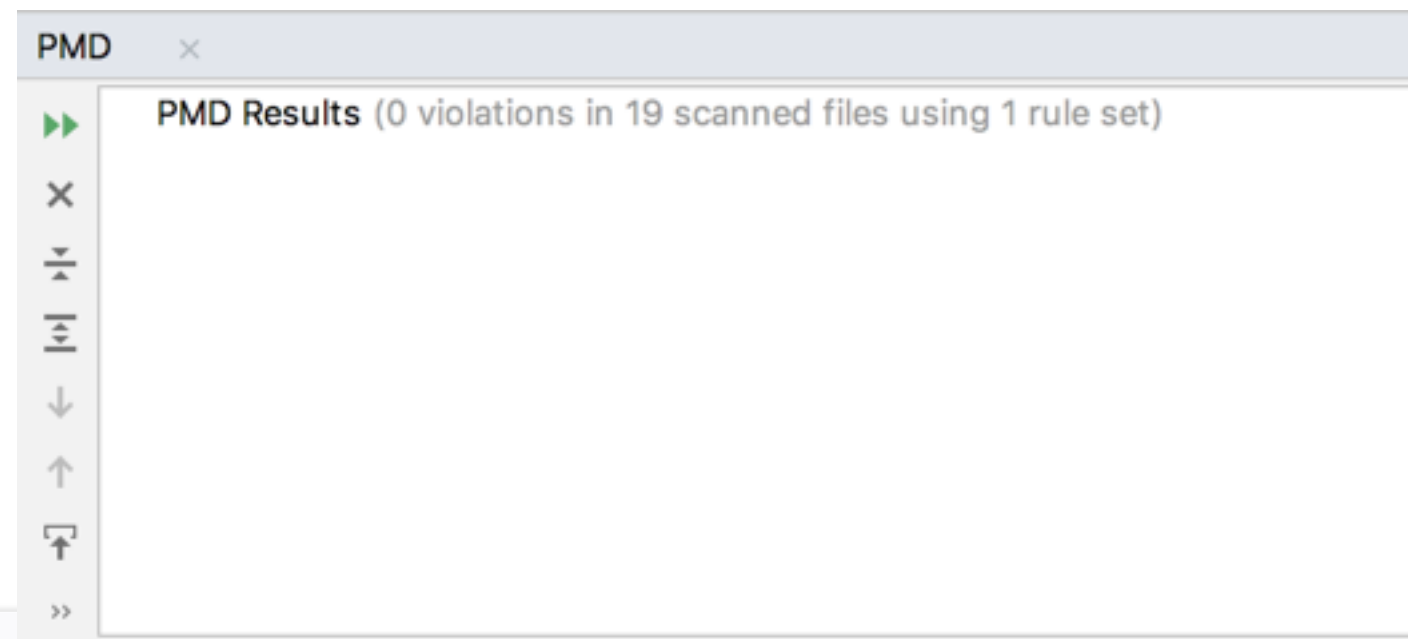


STATIC ANALYSIS RESULTS: PMD

Backend

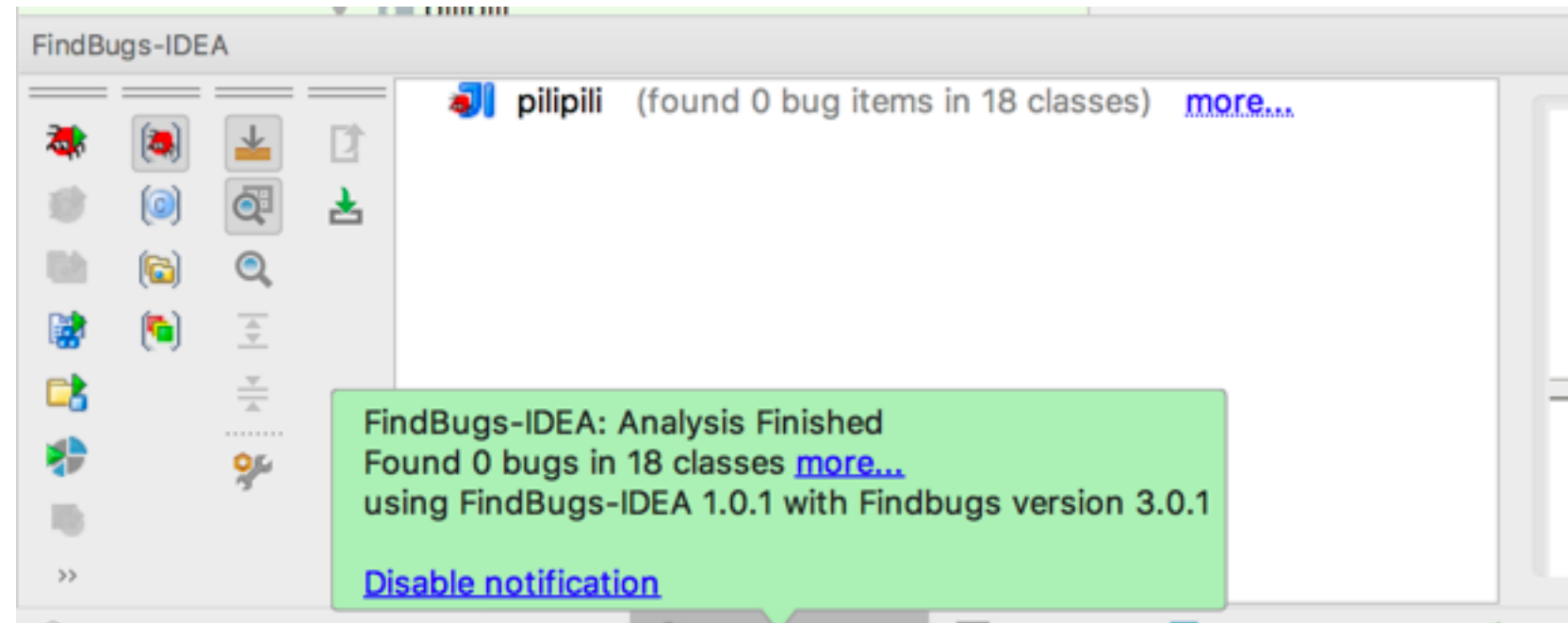


Frontend

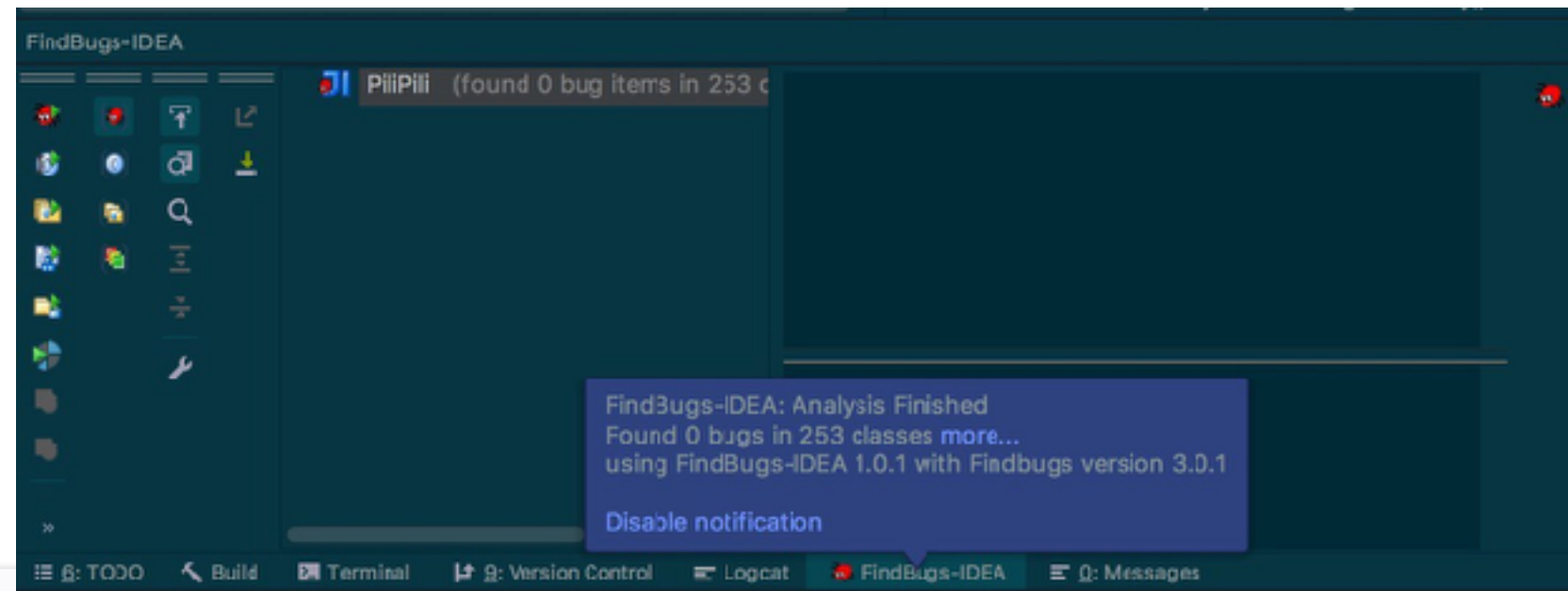


STATIC ANALYSIS RESULTS: FindBugs

Backend



Frontend



MAIN USAGE: Scenario

Scenario 1:

When a user wants to add fancy filter to his/her photo.

Scenario 2:

When a user wants to transfer image style to his/her photo.

Scenario 3:

When a user wants to share photo with his/her friends.

Scenario 4:

When a user wants to have a private place to save his/her photo.

CONCLUSIONS

- UI : Material Design and Uniform Style

- Development Technique

App

Glide: fast and efficient open source framework

Retrofit: Type-safe HTTP client for Android and Java

GPUImage: GPU-accelerated filters and other effects to images

Tensorflow: Open source machine learning framework

Backend

Springboot + Hibernate + MySQL

- Test Technique

ESpresso + JUnit + MOCKMVC

FUTURE WORK

