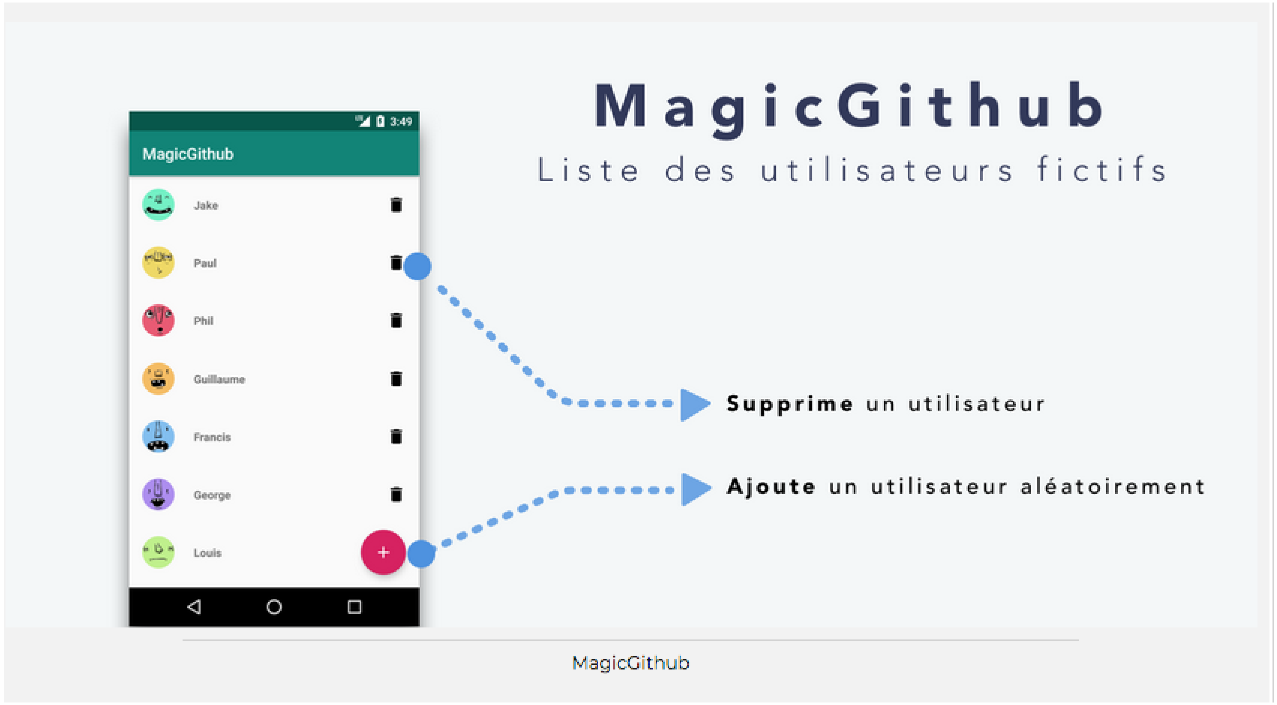
Projet 2 :

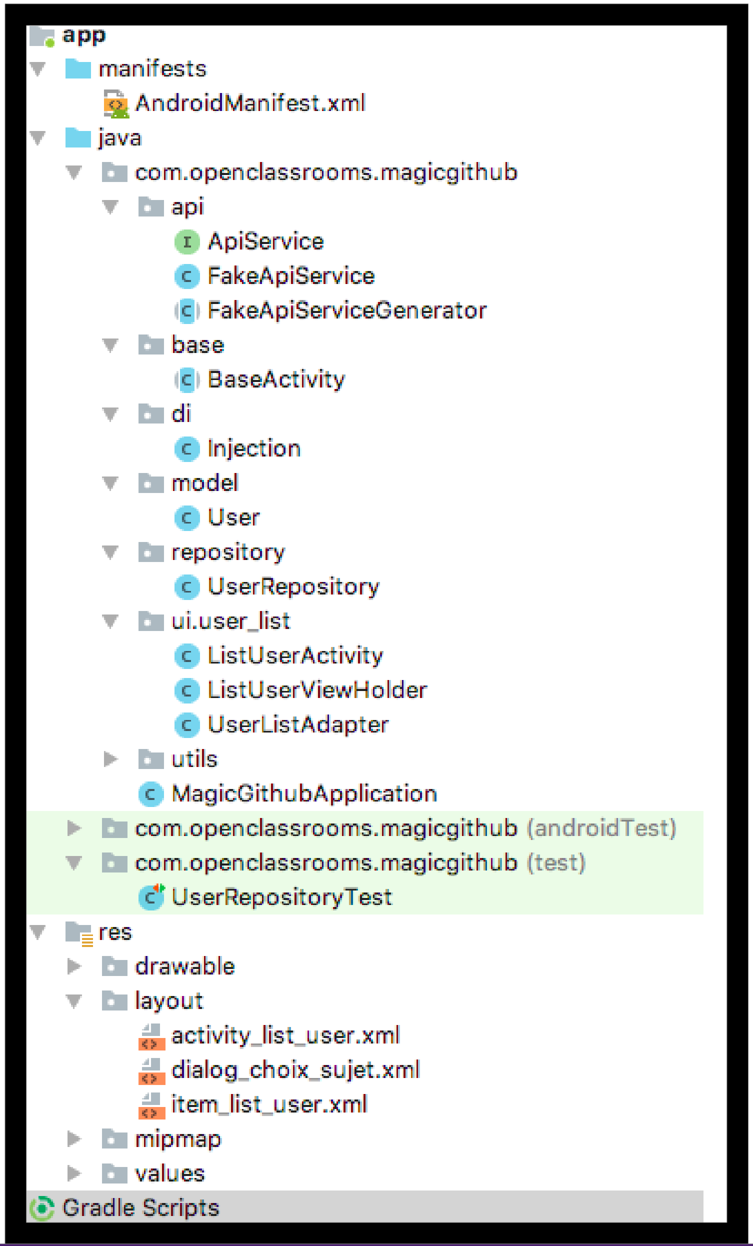
-Analysez les tests de l'application Magic GitHub



Projet 2 :

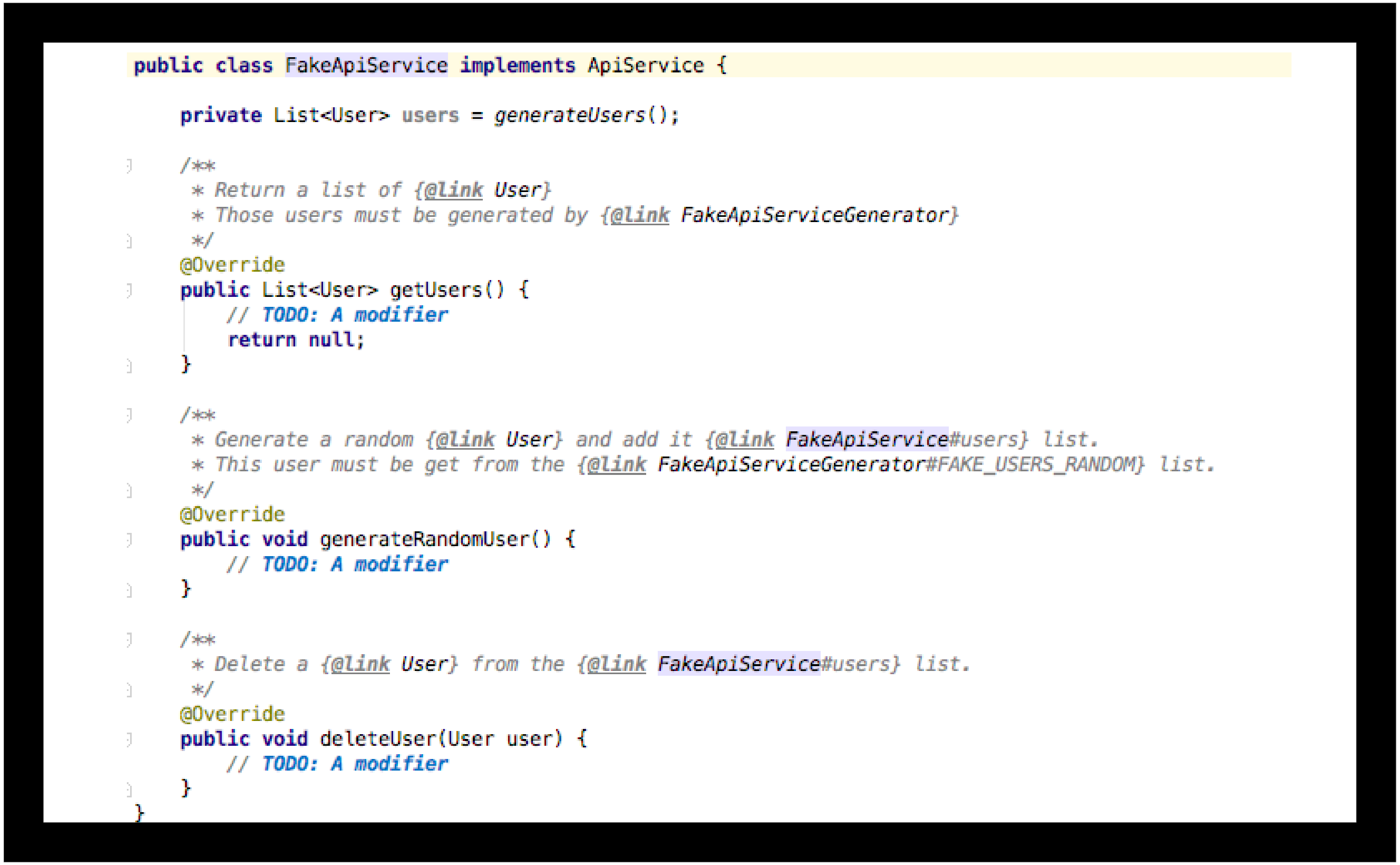
-Analysez les tests de l’application Magic GitHub

Architecture du projet



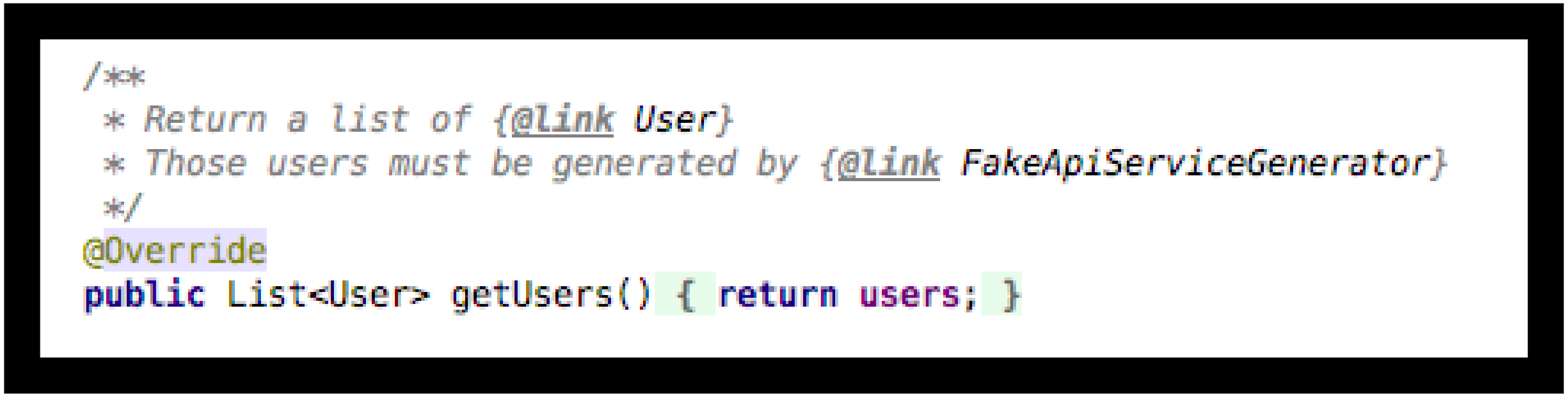
```
app
▼ manifests
    AndroidManifest.xml
▼ java
  ▼ com.openclassrooms.magicgithub
    ▼ api
        ApiService
        FakeApiService
        FakeApiServiceGenerator
    ▼ base
        BaseActivity
    ▼ di
        Injection
    ▼ model
        User
    ▼ repository
        UserRepository
    ▼ ui.user_list
        ListUserActivity
        ListUserViewHolder
        UserListAdapter
    ▶ utils
      MagicGithubApplication
  ▶ com.openclassrooms.magicgithub (androidTest)
  ▼ com.openclassrooms.magicgithub (test)
      UserRepositoryTest
▼ res
  ▶ drawable
  ▼ layout
      activity_list_user.xml
      dialog_choix_sujet.xml
      item_list_user.xml
  ▶ mipmap
  ▶ values
Gradle Scripts
```

FakeApiService : non complété

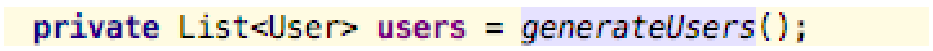


```
public class FakeApiService implements ApiService {

    private List<User> users = generateUsers();

    /**
     * Return a list of {@link User}
     * Those users must be generated by {@link FakeApiServiceGenerator}
     */
    @Override
    public List<User> getUsers() {
        // TODO: A modifier
        return null;
    }

    /**
     * Generate a random {@link User} and add it {@link FakeApiService#users} list.
     * This user must be get from the {@link FakeApiServiceGenerator#FAKE_USERS_RANDOM} list.
     */
    @Override
    public void generateRandomUser() {
        // TODO: A modifier
    }

    /**
     * Delete a {@link User} from the {@link FakeApiService#users} list.
     */
    @Override
    public void deleteUser(User user) {
        // TODO: A modifier
    }
}
```

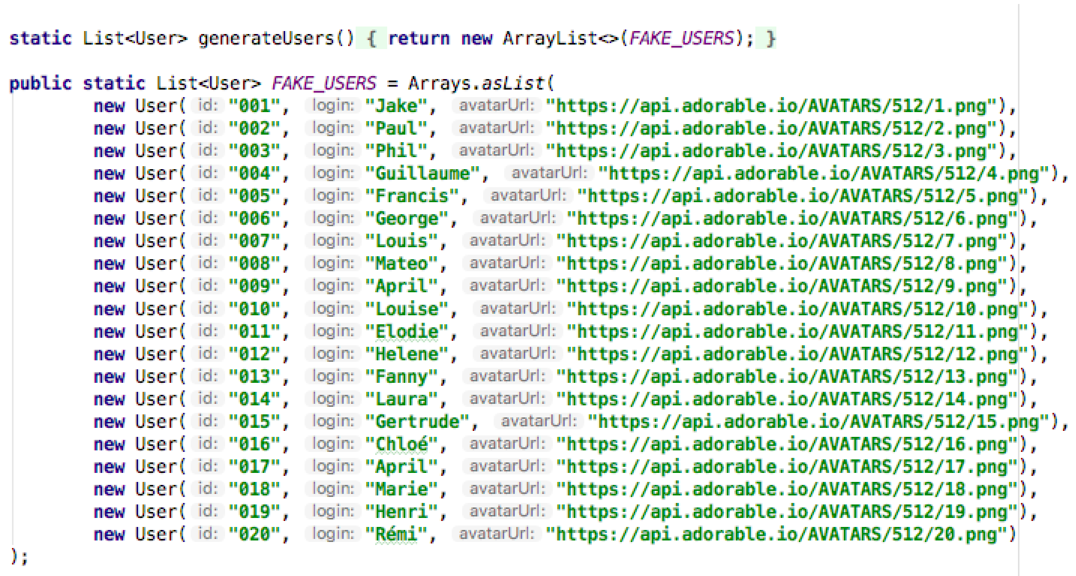
GetUsers() : pour obtenir la liste des utlisateurs



```java
/**
 * Return a list of {@link User}
 * Those users must be generated by {@link FakeApiServiceGenerator}
 */
@Override
public List<User> getUsers() { return users; }
```

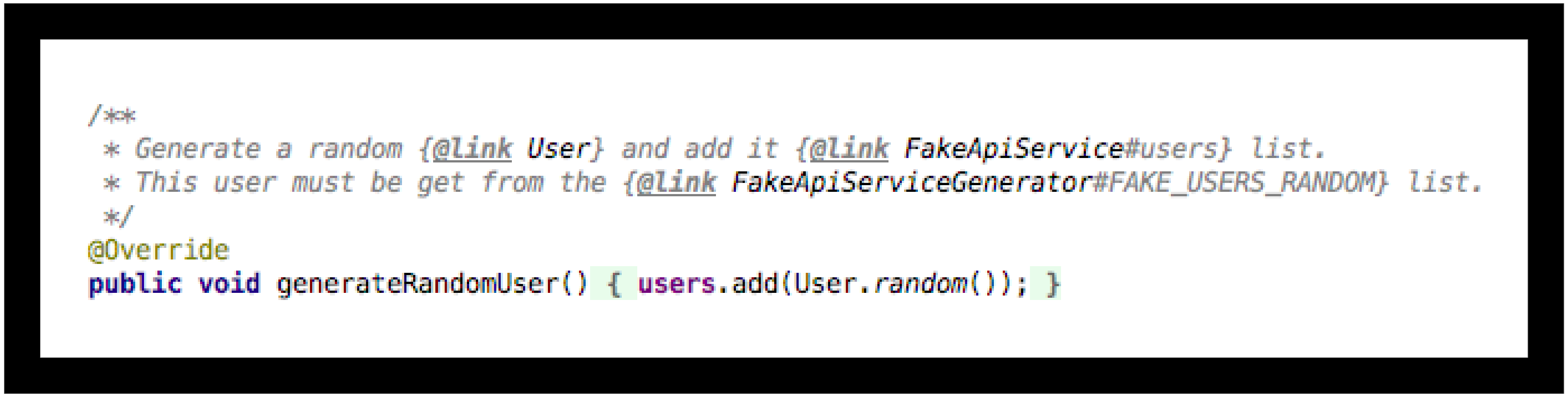


```java
private List<User> users = generateUsers();
```

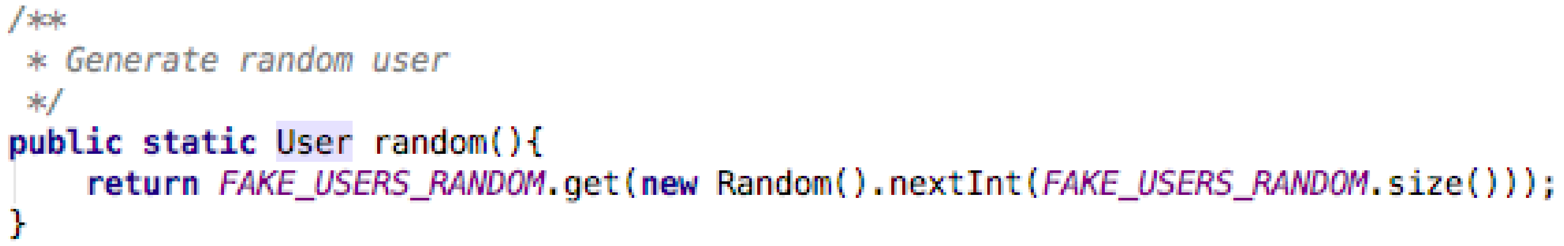


```java
static List<User> generateUsers() { return new ArrayList<>(FAKE_USERS); }

public static List<User> FAKE_USERS = Arrays.asList(
        new User( id: "001",  login: "Jake",  avatarUrl: "https://api.adorable.io/AVATARS/512/1.png"),
        new User( id: "002",  login: "Paul",  avatarUrl: "https://api.adorable.io/AVATARS/512/2.png"),
        new User( id: "003",  login: "Phil",  avatarUrl: "https://api.adorable.io/AVATARS/512/3.png"),
        new User( id: "004",  login: "Guillaume",  avatarUrl: "https://api.adorable.io/AVATARS/512/4.png"),
        new User( id: "005",  login: "Francis",  avatarUrl: "https://api.adorable.io/AVATARS/512/5.png"),
        new User( id: "006",  login: "George",  avatarUrl: "https://api.adorable.io/AVATARS/512/6.png"),
        new User( id: "007",  login: "Louis",  avatarUrl: "https://api.adorable.io/AVATARS/512/7.png"),
        new User( id: "008",  login: "Mateo",  avatarUrl: "https://api.adorable.io/AVATARS/512/8.png"),
        new User( id: "009",  login: "April",  avatarUrl: "https://api.adorable.io/AVATARS/512/9.png"),
        new User( id: "010",  login: "Louise",  avatarUrl: "https://api.adorable.io/AVATARS/512/10.png"),
        new User( id: "011",  login: "Elodie",  avatarUrl: "https://api.adorable.io/AVATARS/512/11.png"),
        new User( id: "012",  login: "Helene",  avatarUrl: "https://api.adorable.io/AVATARS/512/12.png"),
        new User( id: "013",  login: "Fanny",  avatarUrl: "https://api.adorable.io/AVATARS/512/13.png"),
        new User( id: "014",  login: "Laura",  avatarUrl: "https://api.adorable.io/AVATARS/512/14.png"),
        new User( id: "015",  login: "Gertrude",  avatarUrl: "https://api.adorable.io/AVATARS/512/15.png"),
        new User( id: "016",  login: "Chloé",  avatarUrl: "https://api.adorable.io/AVATARS/512/16.png"),
        new User( id: "017",  login: "April",  avatarUrl: "https://api.adorable.io/AVATARS/512/17.png"),
        new User( id: "018",  login: "Marie",  avatarUrl: "https://api.adorable.io/AVATARS/512/18.png"),
        new User( id: "019",  login: "Henri",  avatarUrl: "https://api.adorable.io/AVATARS/512/19.png"),
        new User( id: "020",  login: "Rémi",  avatarUrl: "https://api.adorable.io/AVATARS/512/20.png")
);
```

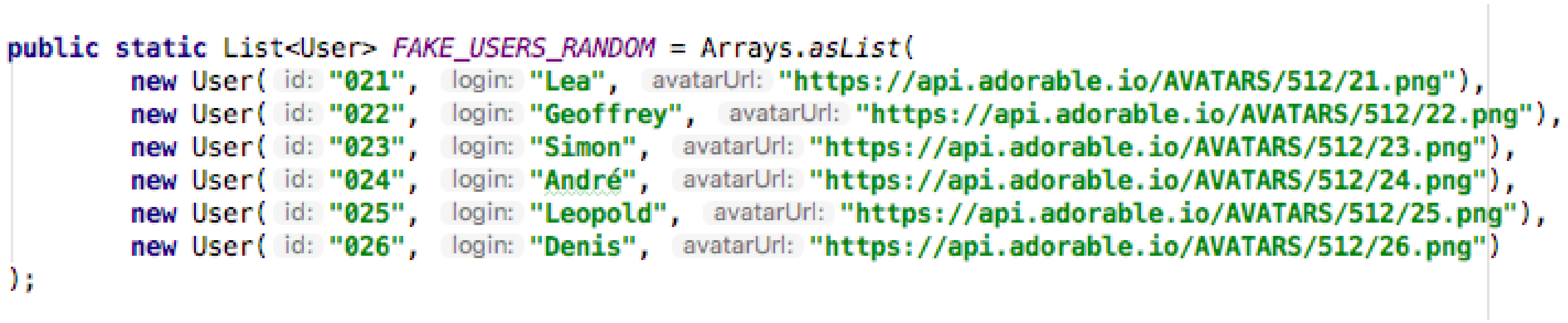
GenerateRandomUser() : pour ajouter un utlisateur au hasard de la liste FAKE_USER_RANDOM



```
/**
 * Generate a random {@link User} and add it {@link FakeApiService#users} list.
 * This user must be get from the {@link FakeApiServiceGenerator#FAKE_USERS_RANDOM} list.
 */
@Override
public void generateRandomUser() { users.add(User.random()); }
```

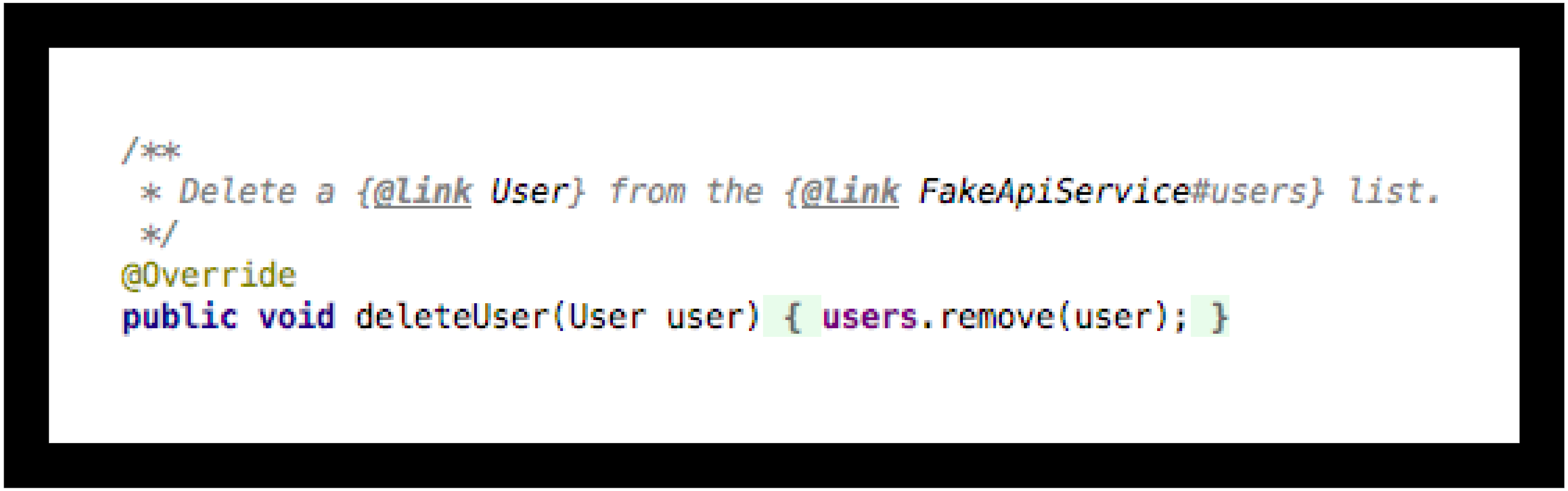


```
/**
 * Generate random user
 */
public static User random(){
    return FAKE_USERS_RANDOM.get(new Random().nextInt(FAKE_USERS_RANDOM.size()));
}
```



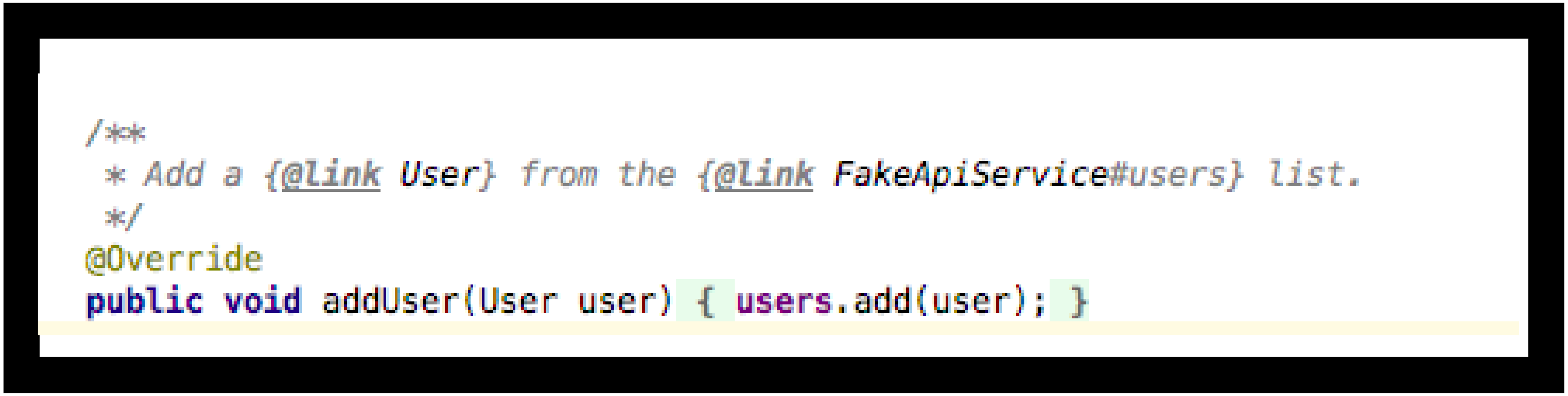
```
public static List<User> FAKE_USERS_RANDOM = Arrays.asList(
        new User( id: "021",  login: "Lea",  avatarUrl: "https://api.adorable.io/AVATARS/512/21.png"),
        new User( id: "022",  login: "Geoffrey",  avatarUrl: "https://api.adorable.io/AVATARS/512/22.png"),
        new User( id: "023",  login: "Simon",  avatarUrl: "https://api.adorable.io/AVATARS/512/23.png"),
        new User( id: "024",  login: "André",  avatarUrl: "https://api.adorable.io/AVATARS/512/24.png"),
        new User( id: "025",  login: "Leopold",  avatarUrl: "https://api.adorable.io/AVATARS/512/25.png"),
        new User( id: "026",  login: "Denis",  avatarUrl: "https://api.adorable.io/AVATARS/512/26.png")
);
```

DeleteUser() Pour supprimer un utilisateur



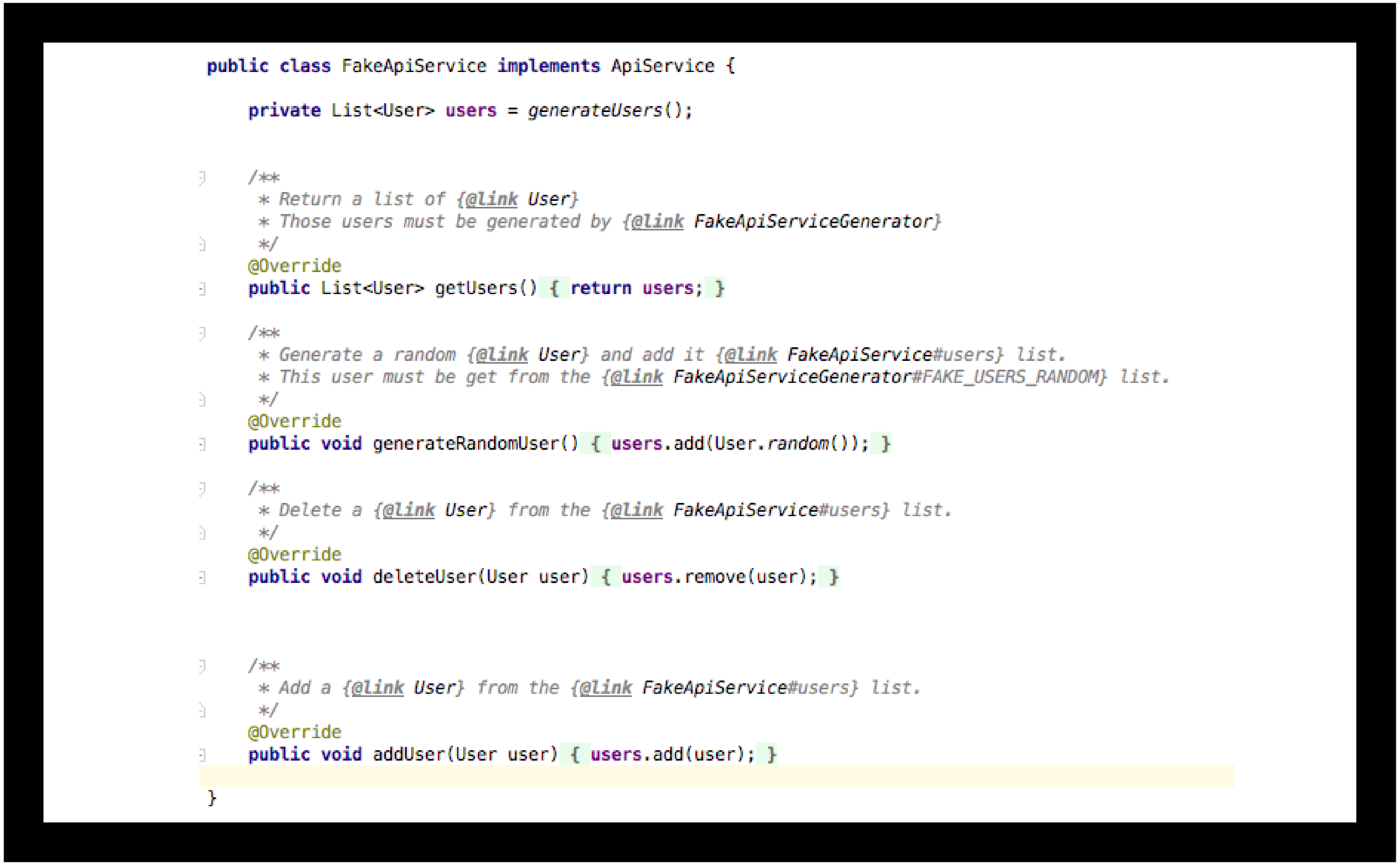
```
/**
 * Delete a {@link User} from the {@link FakeApiService#users} list.
 */
@Override
public void deleteUser(User user) { users.remove(user); }
```

AddUsers() : Pour ajouter un utilisateur (fonction non demandée qui permettra à l'utilisateur de l'application de choisir le nom du User ajouté)



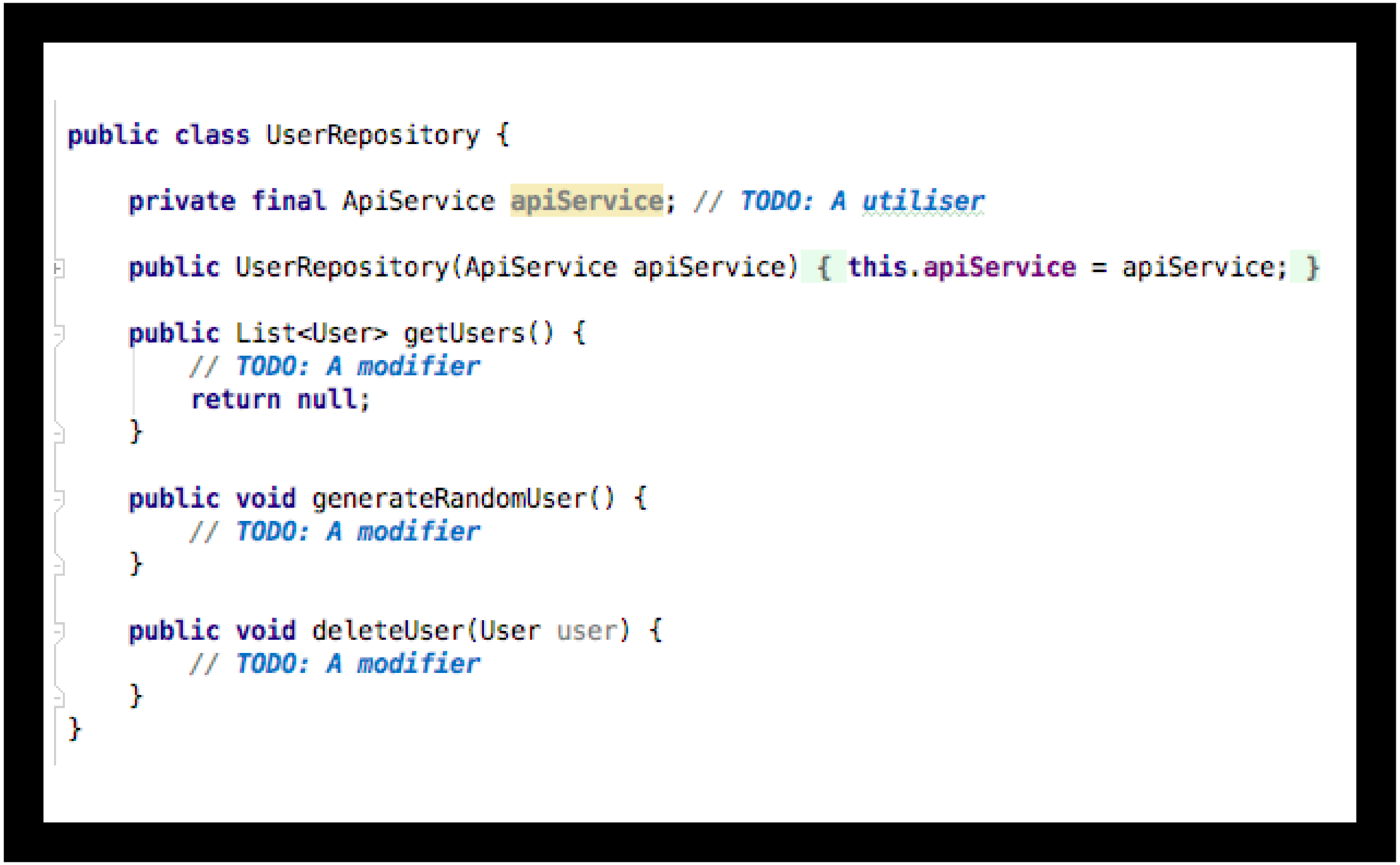
```java
/**
 * Add a {@link User} from the {@link FakeApiService#users} list.
 */
@Override
public void addUser(User user) { users.add(user); }
```

FakeApiService : complété



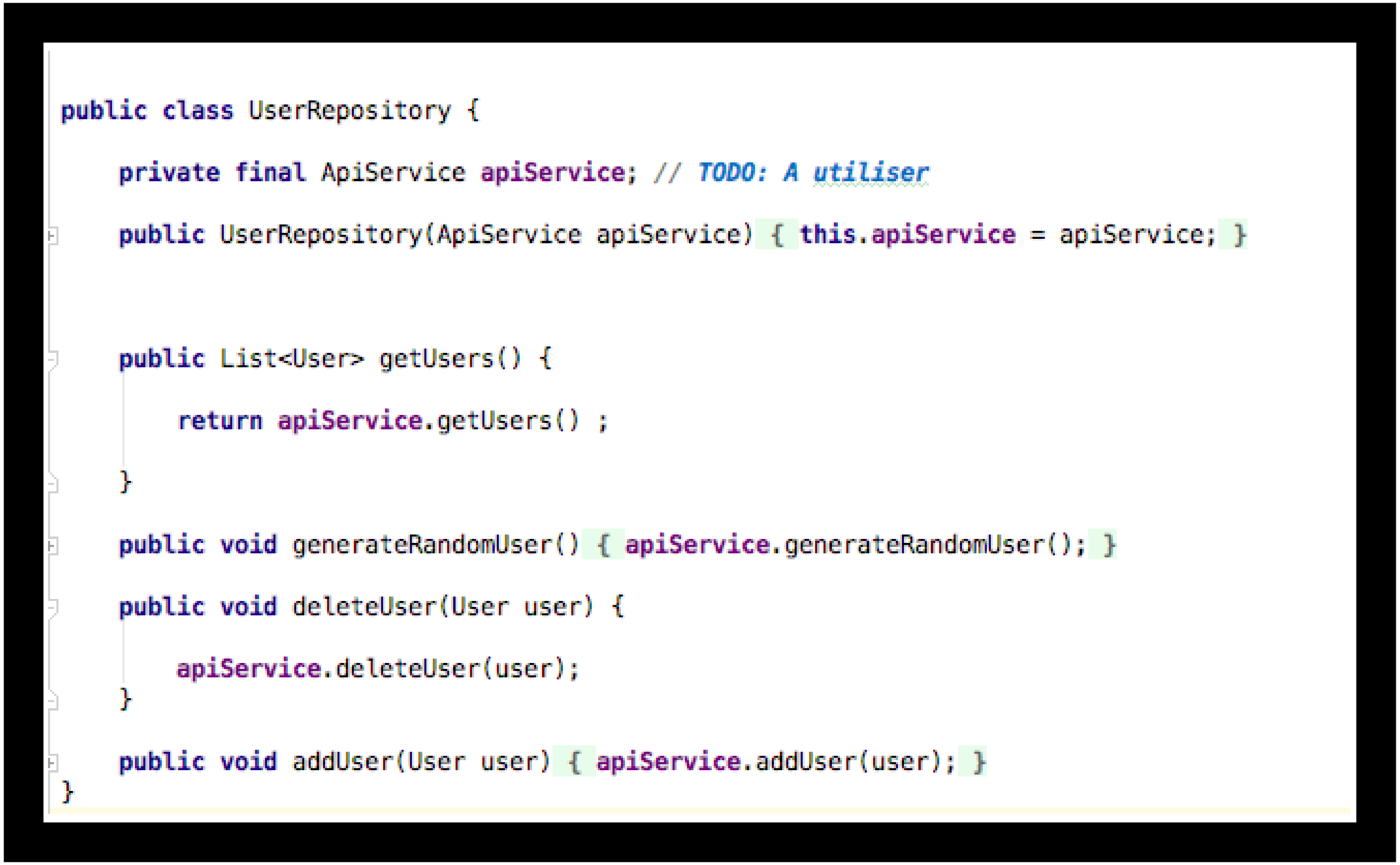
```java
public class FakeApiService implements ApiService {

    private List<User> users = generateUsers();


    /**
     * Return a list of {@link User}
     * Those users must be generated by {@link FakeApiServiceGenerator}
     */
    @Override
    public List<User> getUsers() { return users; }

    /**
     * Generate a random {@link User} and add it {@link FakeApiService#users} list.
     * This user must be get from the {@link FakeApiServiceGenerator#FAKE_USERS_RANDOM} list.
     */
    @Override
    public void generateRandomUser() { users.add(User.random()); }

    /**
     * Delete a {@link User} from the {@link FakeApiService#users} list.
     */
    @Override
    public void deleteUser(User user) { users.remove(user); }



    /**
     * Add a {@link User} from the {@link FakeApiService#users} list.
     */
    @Override
    public void addUser(User user) { users.add(user); }

}
```

UserRepository : non complété



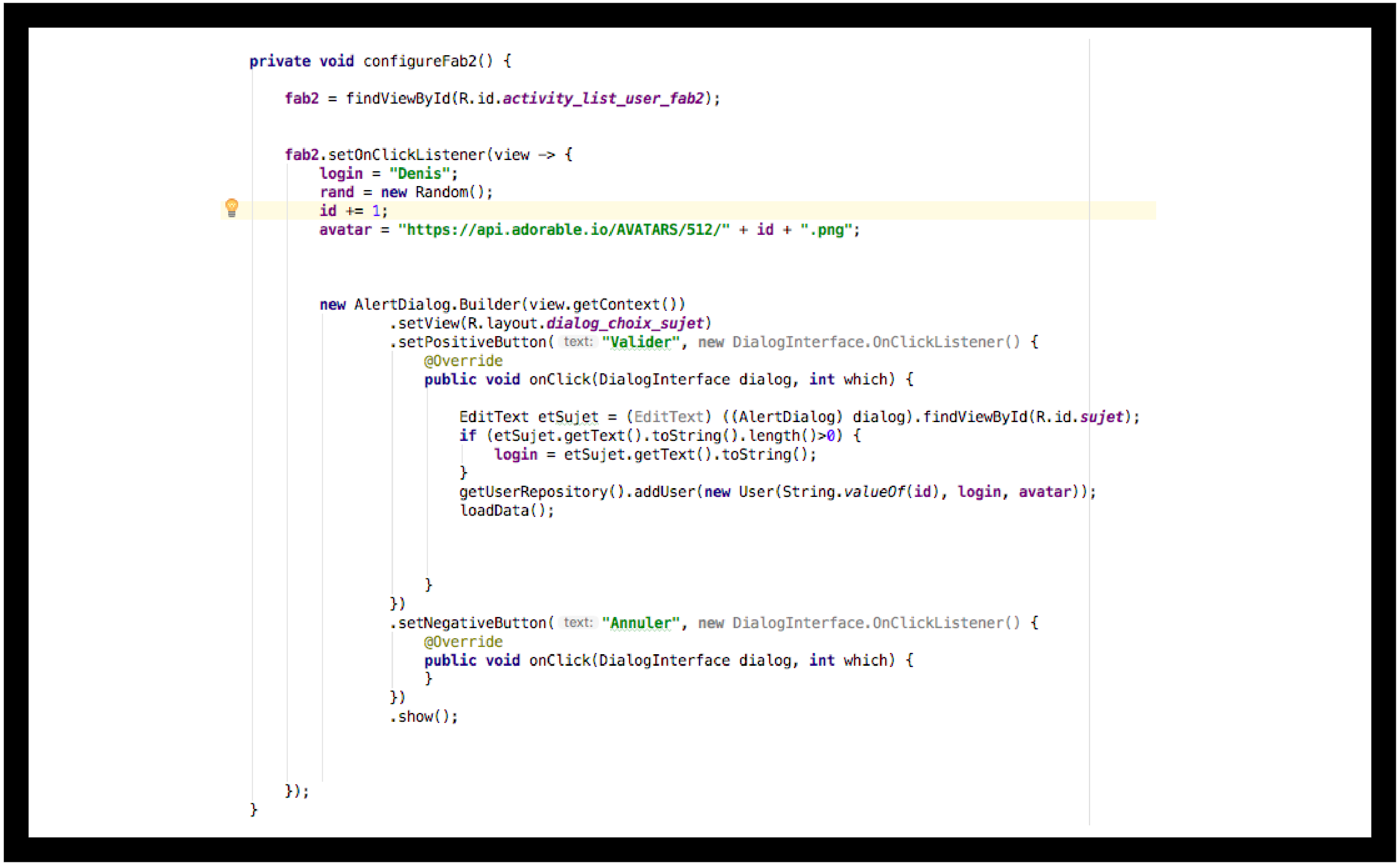
```
public class UserRepository {

    private final ApiService apiService; // TODO: A utiliser

    public UserRepository(ApiService apiService) { this.apiService = apiService; }

    public List<User> getUsers() {
        // TODO: A modifier
        return null;
    }

    public void generateRandomUser() {
        // TODO: A modifier
    }

    public void deleteUser(User user) {
        // TODO: A modifier
    }
}
```

UserRepository : complété



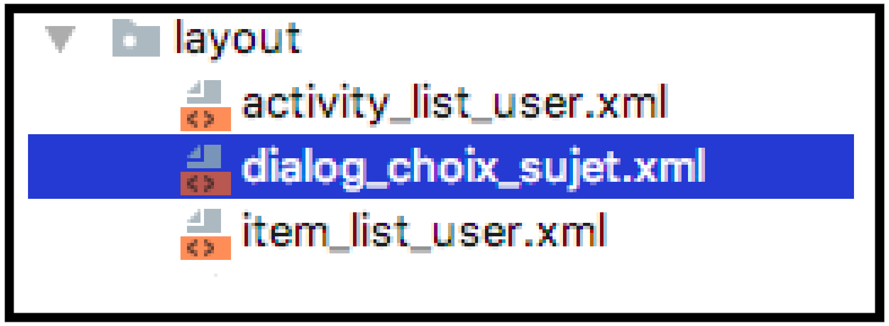
```
public class UserRepository {

    private final ApiService apiService; // TODO: A utiliser

    public UserRepository(ApiService apiService) { this.apiService = apiService; }


    public List<User> getUsers() {

        return apiService.getUsers() ;

    }

    public void generateRandomUser() { apiService.generateRandomUser(); }

    public void deleteUser(User user) {

        apiService.deleteUser(user);
    }

    public void addUser(User user) { apiService.addUser(user); }
}
```

ConfigureFab2() : (fonction non demandée)

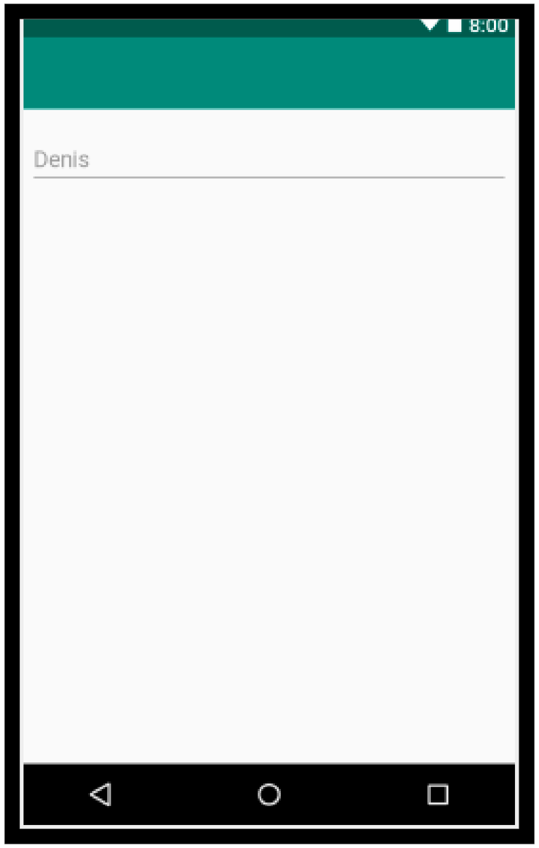


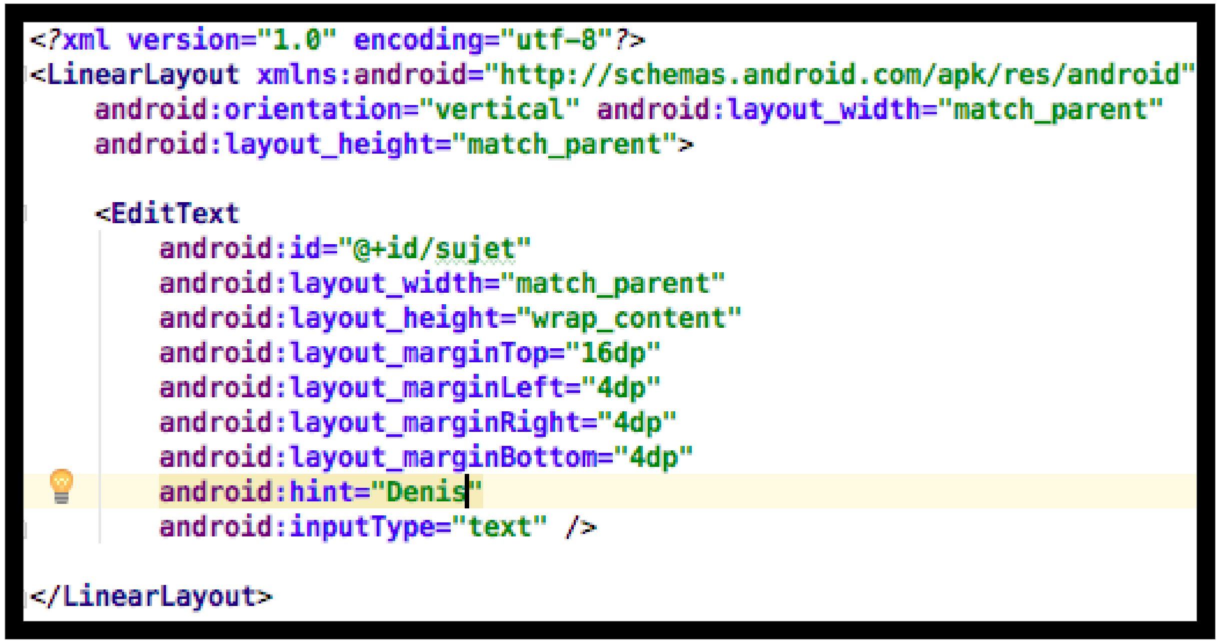
```
private void configureFab2() {

    fab2 = findViewById(R.id.activity_list_user_fab2);


    fab2.setOnClickListener(view -> {
        login = "Denis";
        rand = new Random();
        id += 1;
        avatar = "https://api.adorable.io/AVATARS/512/" + id + ".png";



        new AlertDialog.Builder(view.getContext())
                .setView(R.layout.dialog_choix_sujet)
                .setPositiveButton( text: "Valider", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {

                        EditText etSujet = (EditText) ((AlertDialog) dialog).findViewById(R.id.sujet);
                        if (etSujet.getText().toString().length()>0) {
                            login = etSujet.getText().toString();
                        }
                        getUserRepository().addUser(new User(String.valueOf(id), login, avatar));
                        loadData();



                    }
                })
                .setNegativeButton( text: "Annuler", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                    }
                })
                .show();



    });
}
```

(Layout) dialog-choix-sujet



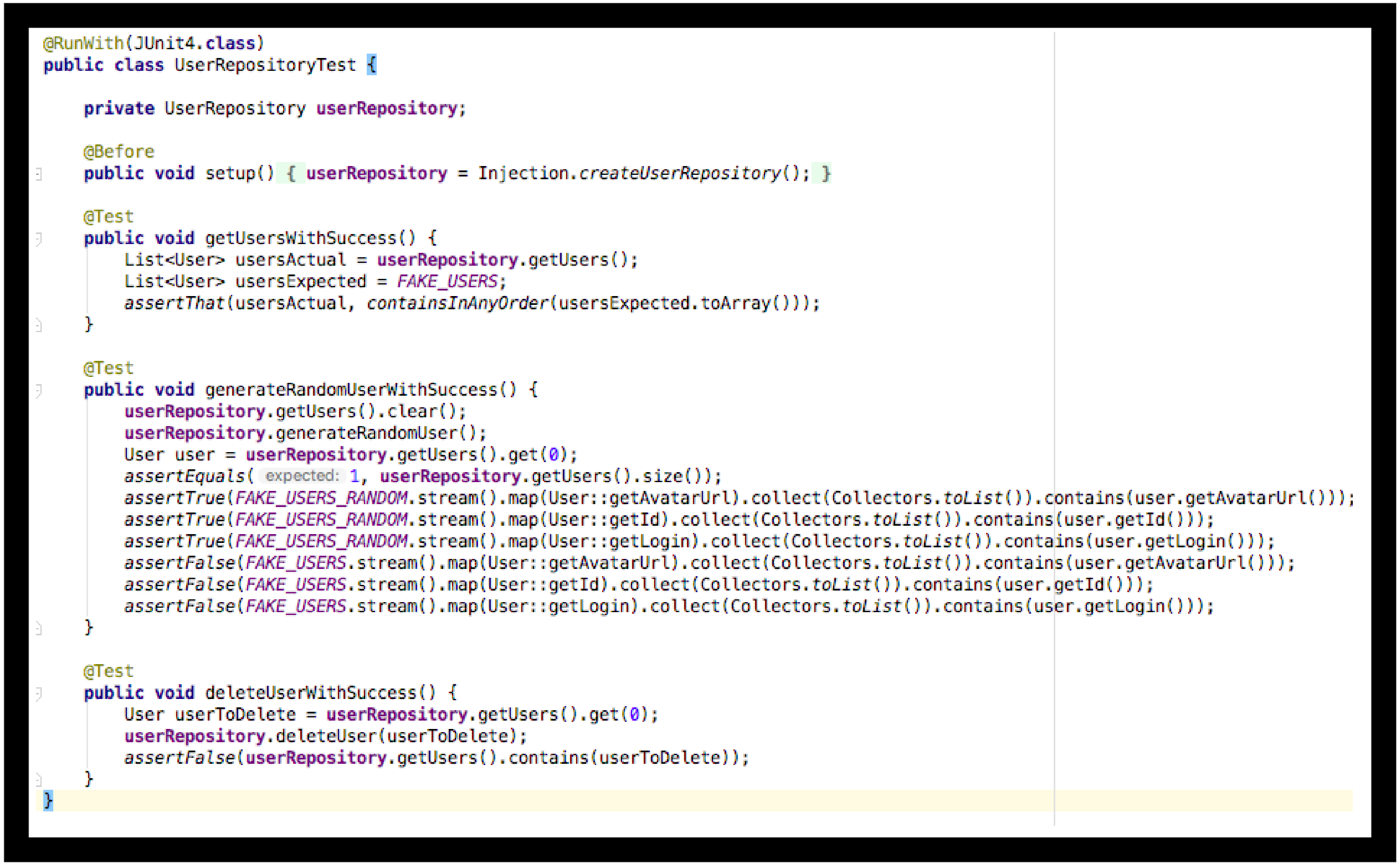
```
layout
    activity_list_user.xml
    dialog_choix_sujet.xml
    item_list_user.xml
```





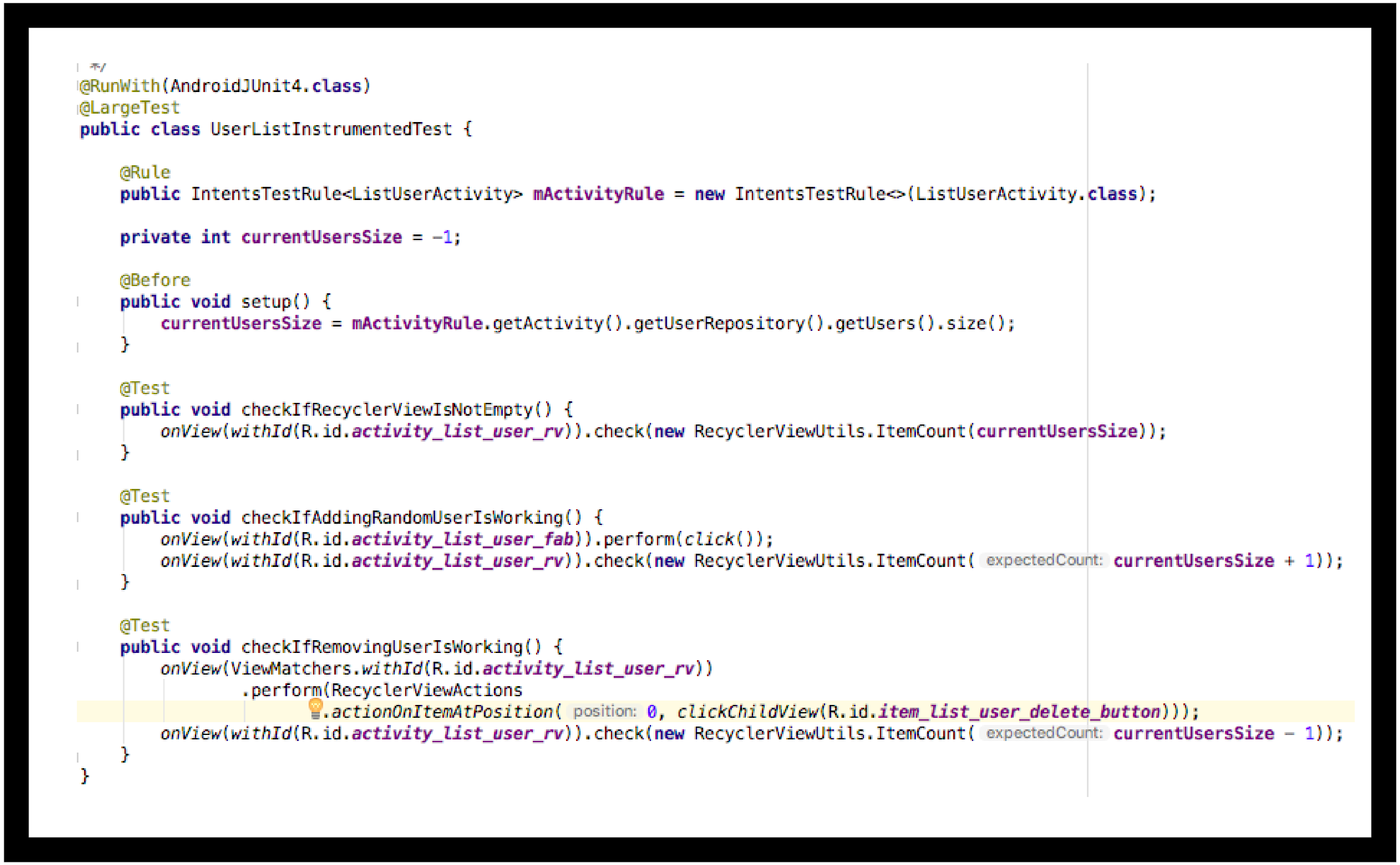
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:id="@+id/sujet"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:layout_marginLeft="4dp"
        android:layout_marginRight="4dp"
        android:layout_marginBottom="4dp"
        android:hint="Denis"
        android:inputType="text" />

</LinearLayout>
```

Les tests unitaires : (rien à modifier)



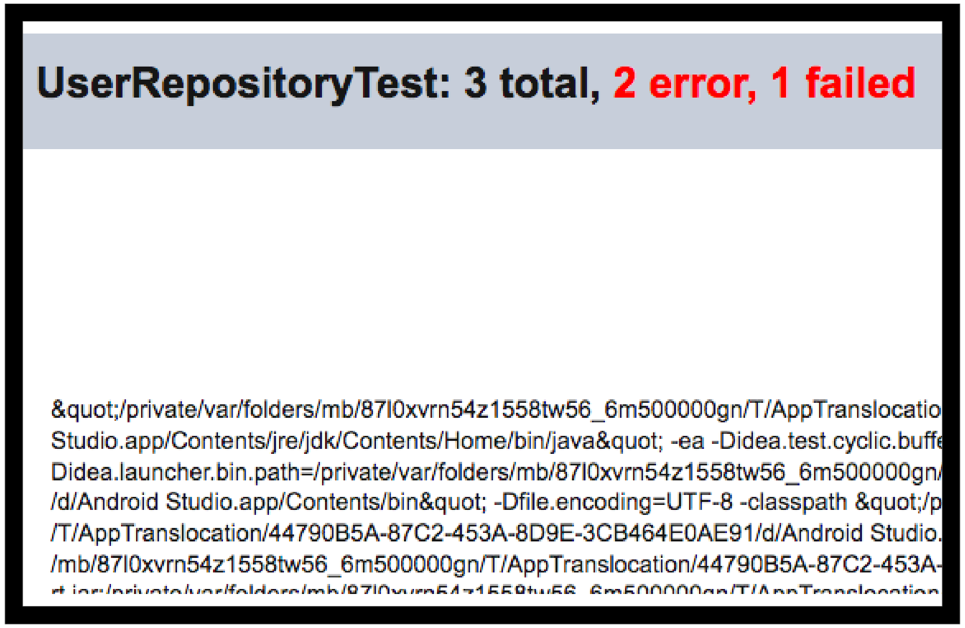
```java
@RunWith(JUnit4.class)
public class UserRepositoryTest {

    private UserRepository userRepository;

    @Before
    public void setup() { userRepository = Injection.createUserRepository(); }

    @Test
    public void getUsersWithSuccess() {
        List<User> usersActual = userRepository.getUsers();
        List<User> usersExpected = FAKE_USERS;
        assertThat(usersActual, containsInAnyOrder(usersExpected.toArray()));
    }

    @Test
    public void generateRandomUserWithSuccess() {
        userRepository.getUsers().clear();
        userRepository.generateRandomUser();
        User user = userRepository.getUsers().get(0);
        assertEquals( expected: 1, userRepository.getUsers().size());
        assertTrue(FAKE_USERS_RANDOM.stream().map(User::getAvatarUrl).collect(Collectors.toList()).contains(user.getAvatarUrl()));
        assertTrue(FAKE_USERS_RANDOM.stream().map(User::getId).collect(Collectors.toList()).contains(user.getId()));
        assertTrue(FAKE_USERS_RANDOM.stream().map(User::getLogin).collect(Collectors.toList()).contains(user.getLogin()));
        assertFalse(FAKE_USERS.stream().map(User::getAvatarUrl).collect(Collectors.toList()).contains(user.getAvatarUrl()));
        assertFalse(FAKE_USERS.stream().map(User::getId).collect(Collectors.toList()).contains(user.getId()));
        assertFalse(FAKE_USERS.stream().map(User::getLogin).collect(Collectors.toList()).contains(user.getLogin()));
    }

    @Test
    public void deleteUserWithSuccess() {
        User userToDelete = userRepository.getUsers().get(0);
        userRepository.deleteUser(userToDelete);
        assertFalse(userRepository.getUsers().contains(userToDelete));
    }
}
```

Les tests instrumentalisés : (rien à modifier)



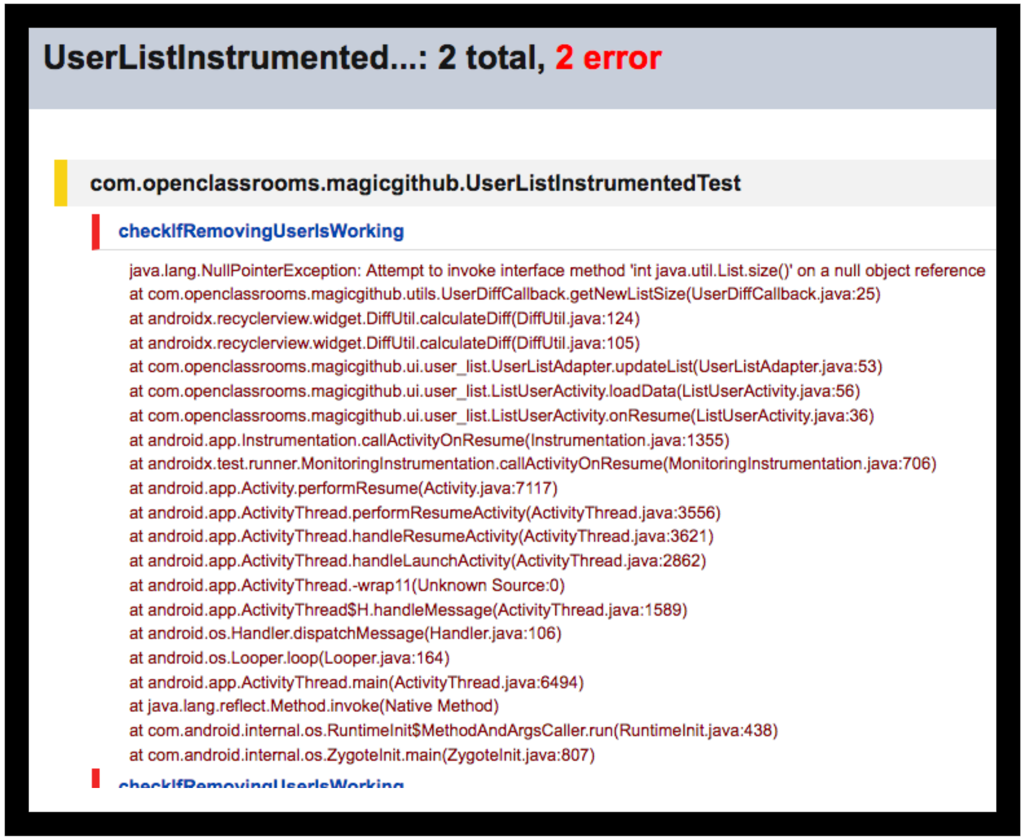
```
 */
@RunWith(AndroidJUnit4.class)
@LargeTest
public class UserListInstrumentedTest {

    @Rule
    public IntentsTestRule<ListUserActivity> mActivityRule = new IntentsTestRule<>(ListUserActivity.class);

    private int currentUsersSize = -1;

    @Before
    public void setup() {
        currentUsersSize = mActivityRule.getActivity().getUserRepository().getUsers().size();
    }

    @Test
    public void checkIfRecyclerViewIsNotEmpty() {
        onView(withId(R.id.activity_list_user_rv)).check(new RecyclerViewUtils.ItemCount(currentUsersSize));
    }

    @Test
    public void checkIfAddingRandomUserIsWorking() {
        onView(withId(R.id.activity_list_user_fab)).perform(click());
        onView(withId(R.id.activity_list_user_rv)).check(new RecyclerViewUtils.ItemCount( expectedCount: currentUsersSize + 1));
    }

    @Test
    public void checkIfRemovingUserIsWorking() {
        onView(ViewMatchers.withId(R.id.activity_list_user_rv))
                .perform(RecyclerViewActions
                        .actionOnItemAtPosition( position: 0, clickChildView(R.id.item_list_user_delete_button)));
        onView(withId(R.id.activity_list_user_rv)).check(new RecyclerViewUtils.ItemCount( expectedCount: currentUsersSize - 1));
    }
}
```

les 2 rapports d'exécution (unitaire et instrumentalisé) des tests finaux qui échouent



UserRepositoryTest: 3 total, 2 error, 1 failed

```
&quot;/private/var/folders/mb/87l0xvrn54z1558tw56_6m500000gn/T/AppTranslocatio
Studio.app/Contents/jre/jdk/Contents/Home/bin/java&quot; -ea -Didea.test.cyclic.buffe
Didea.launcher.bin.path=/private/var/folders/mb/87l0xvrn54z1558tw56_6m500000gn/
/d/Android Studio.app/Contents/bin&quot; -Dfile.encoding=UTF-8 -classpath &quot;/p
/T/AppTranslocation/44790B5A-87C2-453A-8D9E-3CB464E0AE91/d/Android Studio.
/mb/87l0xvrn54z1558tw56_6m500000gn/T/AppTranslocation/44790B5A-87C2-453A-
```



UserListInstrumented...: 2 total, 2 error

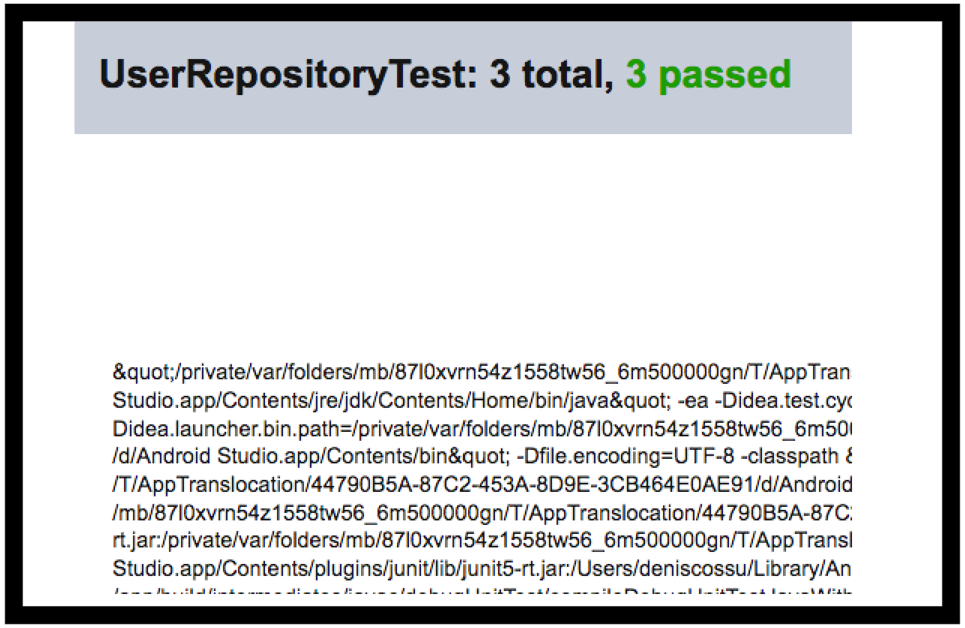
com.openclassrooms.magicgithub.UserListInstrumentedTest

checkIfRemovingUserIsWorking

```
java.lang.NullPointerException: Attempt to invoke interface method 'int java.util.List.size()' on a null object reference
at com.openclassrooms.magicgithub.utils.UserDiffCallback.getNewListSize(UserDiffCallback.java:25)
at androidx.recyclerview.widget.DiffUtil.calculateDiff(DiffUtil.java:124)
at androidx.recyclerview.widget.DiffUtil.calculateDiff(DiffUtil.java:105)
at com.openclassrooms.magicgithub.ui.user_list.UserListAdapter.updateList(UserListAdapter.java:53)
at com.openclassrooms.magicgithub.ui.user_list.ListUserActivity.loadData(ListUserActivity.java:56)
at com.openclassrooms.magicgithub.ui.user_list.ListUserActivity.onResume(ListUserActivity.java:36)
at android.app.Instrumentation.callActivityOnResume(Instrumentation.java:1355)
at androidx.test.runner.MonitoringInstrumentation.callActivityOnResume(MonitoringInstrumentation.java:706)
at android.app.Activity.performResume(Activity.java:7117)
at android.app.ActivityThread.performResumeActivity(ActivityThread.java:3556)
at android.app.ActivityThread.handleResumeActivity(ActivityThread.java:3621)
at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2862)
at android.app.ActivityThread.-wrap11(Unknown Source:0)
at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1589)
at android.os.Handler.dispatchMessage(Handler.java:106)
at android.os.Looper.loop(Looper.java:164)
at android.app.ActivityThread.main(ActivityThread.java:6494)
at java.lang.reflect.Method.invoke(Native Method)
at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:438)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:807)
```

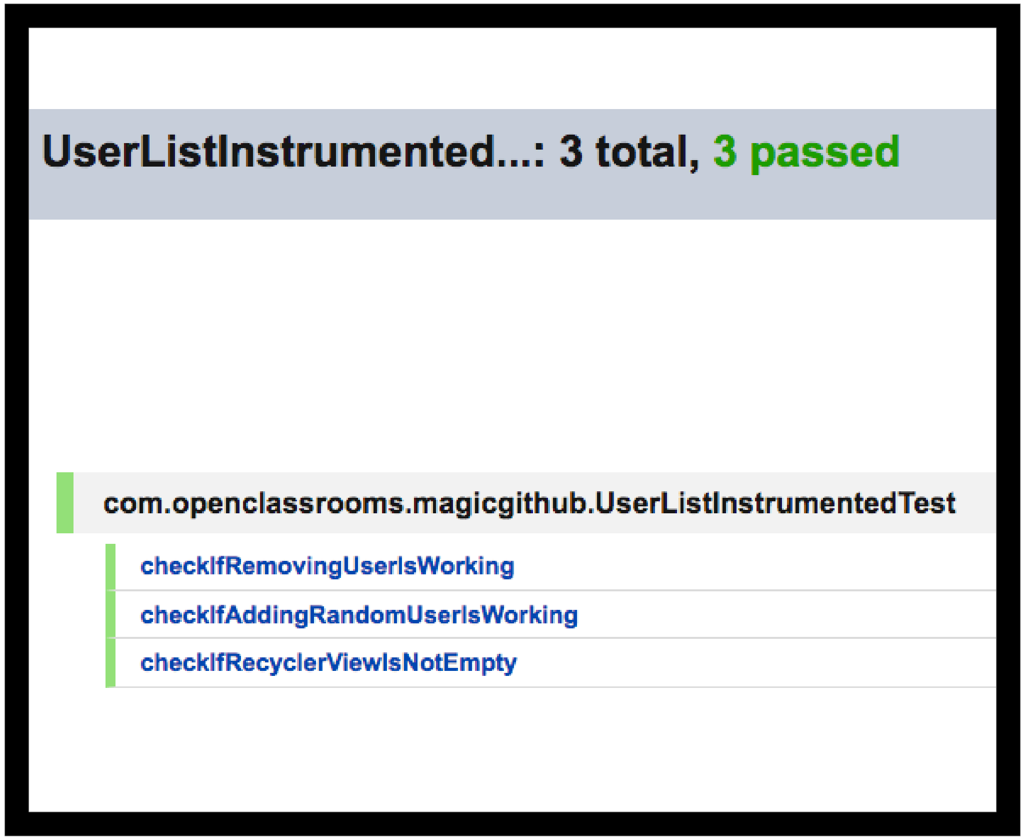
checkIfRemovingUserIsWorking

les 2 rapports d'exécution (unitaire et instrumentalisé) des tests finaux qui réussissent :



UserRepositoryTest: 3 total, 3 passed

```
&quot;/private/var/folders/mb/87l0xvrn54z1558tw56_6m500000gn/T/AppTran
Studio.app/Contents/jre/jdk/Contents/Home/bin/java&quot; -ea -Didea.test.cyc
Didea.launcher.bin.path=/private/var/folders/mb/87l0xvrn54z1558tw56_6m50
/d/Android Studio.app/Contents/bin&quot; -Dfile.encoding=UTF-8 -classpath &
/T/AppTranslocation/44790B5A-87C2-453A-8D9E-3CB464E0AE91/d/Android
/mb/87l0xvrn54z1558tw56_6m500000gn/T/AppTranslocation/44790B5A-87C
rt.jar:/private/var/folders/mb/87l0xvrn54z1558tw56_6m500000gn/T/AppTransl
Studio.app/Contents/plugins/junit/lib/junit5-rt.jar:/Users/deniscossu/Library/An
```



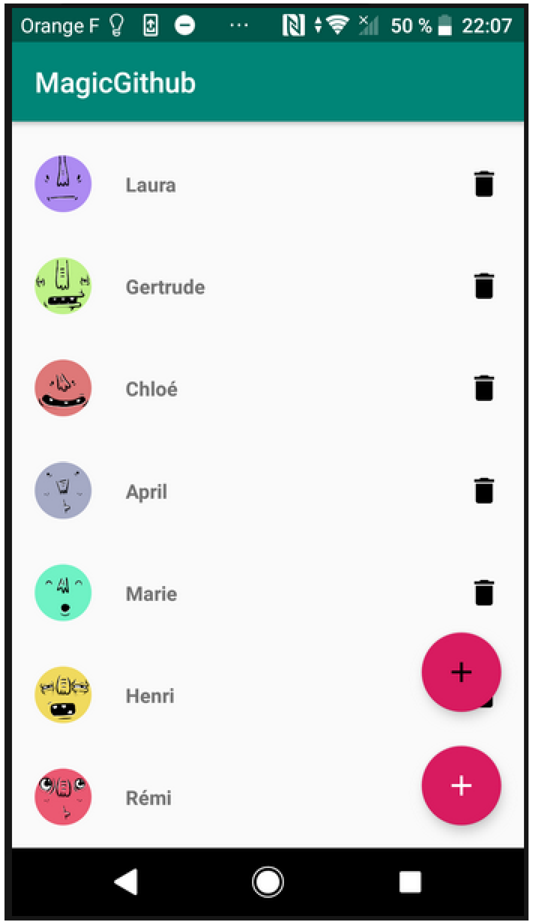
UserListInstrumented...: 3 total, 3 passed

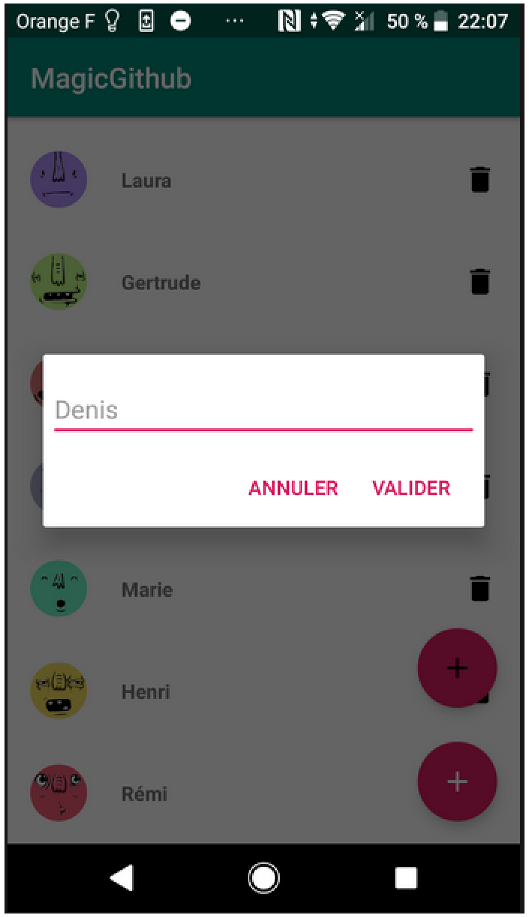
com.openclassrooms.magicgithub.UserListInstrumentedTest

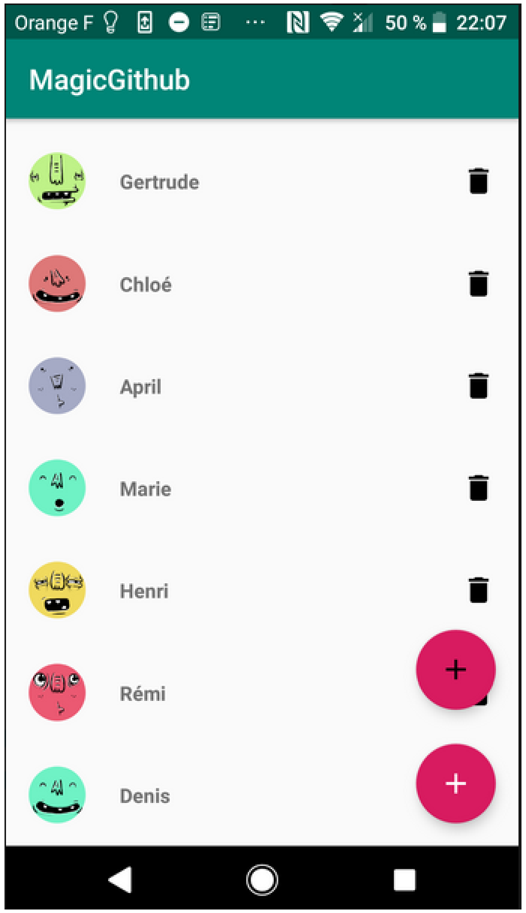
- checkIfRemovingUserIsWorking
- checkIfAddingRandomUserIsWorking
- checkIfRecyclerViewIsNotEmpty

Parcours Développeur Android

Conclusion : Démonstration à l'aide de ApowerMirror







Cossu Denis -Projet 2-