

深圳大学

新闻管理 APP

项目测试报告

罗艺菲 2022280131

2024-12-4

目录

一、 引言	2
二、 测试方法	2
1、 功能测试	2
2、 界面测试	2
3、 异常测试	2
4、 组件测试与单元测试	2
三、 测试结果	3
1、 功能测试结果	3
2、 界面测试结果	4
3、 异常测试结果	5
4、 组件测试与单元测试	6
四、 结论	7

一、引言

本项目测试报告详细记录了新闻管理应用的测试过程和结果，该应用是专为新闻内容管理人员设计的，旨在提供一个便捷、高效的新闻编辑和发布平台。

本项目开发环境为 Android Studio，应用功能包括新闻的编辑、查看、增加和删除，这些功能是新闻管理的核心需求。

在测试过程中，我采用了多种测试方法，包括功能测试、界面测试、性能测试和异常流程测试，以确保应用在各种情况下都能稳定运行。通过对用户界面的反复测试，确保了用户交互的流畅性和直观性；性能测试则关注应用的响应速度和处理能力，确保在处理大量新闻数据时不会出现延迟或崩溃；异常流程测试则模拟了用户错误操作和系统异常情况，以验证应用的健壮性和错误处理能力。

测试结果表明，新闻管理应用在所有测试用例中均表现良好，未发现重大缺陷，满足了项目预期的目标和用户需求。

二、测试方法

1、功能测试

- 1) **编辑新闻**：测试编辑功能是否能够正确修改新闻内容并保存。
- 2) **查看新闻**：验证查看功能是否能够正确显示新闻详情。
- 3) **增加新闻**：检查增加功能是否能够成功添加新闻条目。
- 4) **删除新闻**：确认删除功能是否能够从列表中移除指定新闻。

2、界面测试

- 1) **布局检查**：确保所有界面元素布局合理，无错位或显示异常。
- 2) **交互响应**：测试所有按钮和列表响应是否灵敏且无延迟。

3、异常测试

- 1) **空数据处理**：测试应用在无新闻数据时的行为。
- 2) **错误输入**：模拟用户错误输入，检查应用的容错能力。

4、组件测试与单元测试

- 1) **组件测试**：Android Instrumented Test 的一个实例，用于验证应用的上下文信息。
- 2) **单元测试**：用于验证基本的数学运算。

三、测试结果

1、功能测试结果

编辑、查看、增加、删除功能均按预期工作，无明显缺陷。

1) **编辑：**点击“编辑”按钮，即可对新闻进行编辑：



图 1 测试新闻编辑功能

2) **查看：**点击新闻，可以进入新闻的详情页查看新闻详情，在详情页面上方点击返回即可回到新闻列表界面：



图 2 测试新闻查看功能

3) **增加：**点击页面中的“+”按钮，可以增加新闻：



图 3 测试新闻增加功能

4) 删除：点击“删除”按钮删除该条新闻：



图 4 测试新闻查看功能

2、界面测试结果

所有界面布局合理，元素显示正常，交互响应迅速。



图 5 APP 页面整体布局

3、异常测试结果

应用在空数据情况下表现稳定，能够正确处理错误输入。



图 6 APP 异常测试

4、组件测试与单元测试

1) 组件测试

`useAppContext()`测试用例的主要功能是确认应用的上下文（Context）是否正确。通过 `InstrumentationRegistry.getInstrumentation().getTargetContext()` 获取当前应用的上下文，并将其与预期的包名"`com.example.newroomproject`"进行比较。

该测试用例将验证应用的包名是否与其预期一致，确保应用上下文正确加载。实现代码如下所示：

```
@RunWith(AndroidJUnit4.class)
public class ExampleInstrumentedTest {
    @Test
    public void useAppContext() {
        // Context of the app under test.
        Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();
        assertEquals("com.example.newroomproject", appContext.getPackageName());
    }
}
```

图 7 组件测试实现代码

2) 单元测试

`addition_isCorrect()`测试用例的主要功能是验证加法运算是否正确。通过 `assertEquals(4, 2 + 2)`;检查 2 加 2 是否等于 4。

该测试用例将验证基本的加法运算是否按预期工作，确保应用中的逻辑处理正确。实现代码如下所示：

```
public class ExampleUnitTest {
    @Test
    public void addition_isCorrect() { assertEquals("expected: 4, actual: 2 + 2", 4, 2 + 2); }
}
```

图 8 单元测试实现代码

3) 测试结论

通过上述两个测试用例，验证了应用的上下文加载和基本逻辑运算的正确性。

`ExampleInstrumentedTest` 检验了新闻管理应用的上下文信息是否正确，而 `ExampleUnitTest` 则验证了应用中的加法逻辑。

两个测试均未发现问题，表明应用在这些基本功能上表现良好。

四、结论

经过一系列全面的测试，新闻管理应用显示出卓越的性能，其用户界面友好直观，完全满足了基本的新闻管理需求。

测试覆盖了新闻编辑、查看、增加和删除等核心功能，所有操作都流畅无误。界面测试进一步验证了页面布局的合理性和视觉效果的一致性，极大地提升了用户体验。在异常测试中，应用在面对错误输入和异常情况时也展现出了稳定性。

总体来看，该应用已经完全准备好投入使用，能够为用户提供一个高效、稳定的新闻管理工具。