

测试文档

一、功能性测试（黑盒测试）

在我们小组实现的 Android 应用中，主要有以下几个功能：按课程名搜索图书；按图书名搜索图书；扫条形码查询图书；获取图书详细信息，以及图书预约。

在这次的测试中，我们主要围绕了按课程名搜索图书、按图书名搜索图书、扫条形码查询图书这三个功能进行了黑盒测试。因为这四个功能是我们应用的核心用例实现，所以测试的重点也落在这里。

在测试过程中，我们根据这个学期的软件测试技术课程上学习到的黑盒测试技术进行功能性测试。这次我们使用的技术主要是应用等价类划分选择测试用例的测试技术。该技术主要按照以下几个步骤进行：

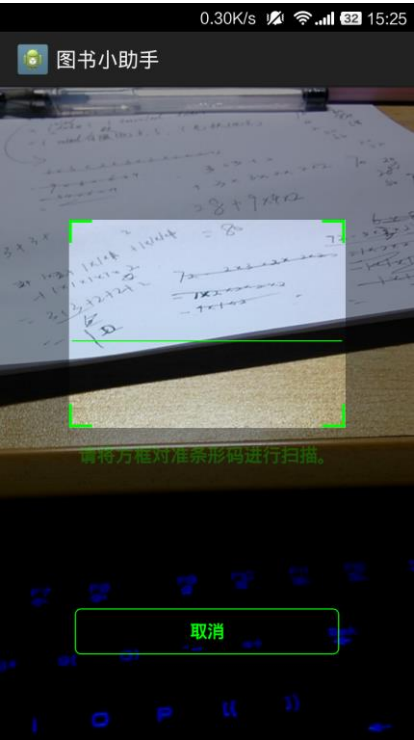
(1) Identify the input domain

在应用中，对于以上的三个功能点，有如下的界面和输入设计：

按课程名搜索和按图书名搜索图书：



扫条形码搜索图书:



根据上面的功能界面，对于每个功能点，我们能确定其输入域:

按课程名搜索图书:

Variable	Type	Value(s)
course_name	字符串	课程名

按图书名搜索图书:

Variable	Type	Value(s)
book_name	字符串	图书名

扫条形码搜索图书:

Variable	Type	Value(s)
1D barcode	图像	图书名

那么，我们可以得知，分别对于每个功能，其输入域有如下关系:

按课程名搜索图书:

$Input\ Domain \subseteq S = course_name$

按图书名搜索图书：

$$Input\ Domain \subseteq S = book_name$$

扫条形码搜索图书：

$$Input\ Domain \subseteq S = 1Dbarcode$$

(2) Equivalence classing

接下来，我们划分各个变量对应的等价类：

Variable	Partition
course_name	{{}, {valid_course}, {invalid_course}}

Variable	Partition
book_name	{{}, {valid_bookname}, {invalid_bookname}}

Variable	Partition
1D barcode	{{valid_barcode}, {invalid_barcode}}

备注：

1. {}表示空输入，即不输入任何内容；

(3) Combine equivalence classes

根据上面步骤中，对每个变量划分的对应的等价类，现在我们可以组合各个单等价类，来得到各个功能的输入域的等价类划分。又因为组合过程中存在不可测的等价类，一般说来还需要进行去除不可测等价类的步骤。但是，在这里我们进行简化和整合，在组合等价类的过程中，就考虑了不可测等价类的情况。具体等价类划分如下：

按课程名搜索图书：

可能的等价类：3 种，实例如下：

{{{}}}
{{(valid_course)}}

{{(invalid_course)}}

按图书名搜索图书：

可能的等价类：3 种，实例如下：

{{({})}}

{{(valid_bookname)}}

{{(invalid_bookname)}}

扫条形码搜索图书：

可能的等价类：2 种，实例如下：

{{(valid_barcode)}}

{{(invalid_barcode)}}

(4) Selecting test case

根据上面的等价类划分，为每个等价类划分实例选择测试用例。

按课程名搜索图书：

1. {{({})}}

选择的测试用例为： t1: < {} >

测试结果：



2. {(valid_course)}

选择的测试用例为： t2: < {数据挖掘} >

测试结果：



3. {(invalid_course)}

选择的测试用例为： t3: < {fewatwa} >

测试结果：



按图书名搜索图书：

1. {{{}}

选择的测试用例为： t4: < {} >

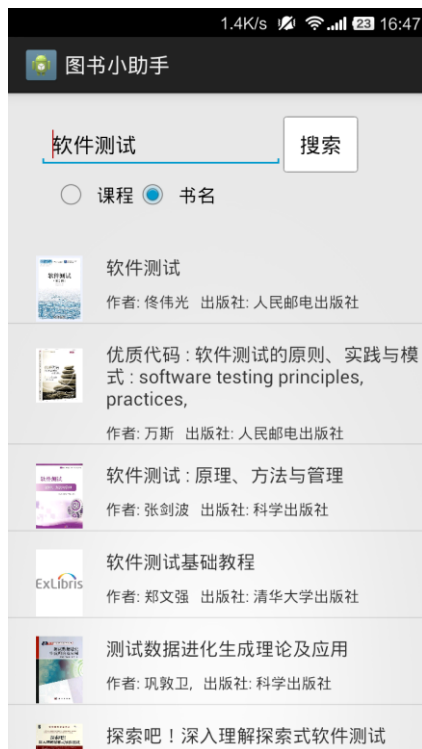
测试结果：



2. {(valid_bookname)}

选择的测试用例为： t5: < {软件测试} >

测试结果：



3. {(invalid_bookname)}

选择的测试用例为： t6: < {fea@123} >

测试结果：



扫条形码搜索图书：

1. {(valid_barcode)}

选择的测试用例：《Python 基础教程（第2版）》的条形码：



测试结果：



2. {(invalid_barcode)}

选择的测试用例：非图书类条形码：



测试结果：



(5) Analysis

根据上面选择的测试用例，对按课程名搜索图书，按图书名搜索图书和扫条形码搜索图书进行了测试，基本覆盖了这三个关键用例，并且测试了应用的正确性，以及处理一些问题输入的错误处理能力。经过测试，应用的功能性实现基本正确。