# Book Store

# Test Cases Document

## Milestone 1

A system setup with catalog, order and front-end server all running on the local machine. A client process calls the front-end server with the following requests.

### Testcase #1

**Scenario**
Request: buy/1 on front-end server. Initial stock of the item is 24.

**Expected Output**
Buy an item with itemNumber 1. The stock of the item should be reduced by 1.

**Output**


```
{
  "msg": "",
  "res": [
    {
      "cost": "25",
      "itemNumber": "1",
      "stock": "23",
      "title": "How to get a good grade in 677 in 20 minutes a day.",
      "topic": "Distributed Systems"
    }
  ]
}
```


### Testcase #2

**Scenario**
Request: lookup/1 on front-end server.

**Expected Output**
Get the item from the database with the item number 1

**Output**

```
Response for the request
{
  "msg": "",
  "res": [
    {
      "cost": "25",
      "itemNumber": "1",
      "stock": "24",
      "title": "How to get a good grade in 677 in 20 minutes a day.",
      "topic": "Distributed Systems"
    }
  ]
}
```

**Testcase #3**

**Scenario**

Request: search/Distributed Systems on front-end server.

**Expected Output**

Get the items from the database with topic Distributed Systems.

**Output**

```
Response for the request
{
  "msg": "",
  "res": [
    {
      "cost": "25",
      "itemNumber": "1",
      "stock": "24",
      "title": "How to get a good grade in 677 in 20 minutes a day.",
      "topic": "Distributed Systems"
    },
    {
      "cost": "15",
      "itemNumber": "2",
      "stock": "15",
      "title": "RPCs for Dummies",
      "topic": "Distributed Systems"
    }
  ]
}
```

**Testcase #4**

**Scenario**
Request: updateCost/1/25 on catalog server

**Expected Output**
The cost of the updated item should be 25

**Output**


```
[
  {
    "itemNumber": "1",
    "title": "How to get a good grade in 677 in 20 minutes a day.",
    "stock": "24",
    "cost": "25",
    "topic": "Distributed Systems"
  }
]
```


**Testcase #5**

**Scenario**
Request: /updateStock/1/25 on catalog server

**Expected Output**
The stock value of the item 1 should be updated to 25

**Output**


```
[
  {
    "itemNumber": "1",
    "title": "How to get a good grade in 677 in 20 minutes a day.",
    "stock": "25",
    "cost": "25",
    "topic": "Distributed Systems"
  }
]
```

## Milestone 2

In a distributed setting, the test.py which contains the tests for the servers take in the Public DNSs for the servers as input and executes the same test cases above and outputs the results in a test.txt file in the local system.

So, in essence, the same test cases get run in a distributed setting and the output for the test cases(which will be similar to the above test cases) will be in the test.txt