

# Test report - version 0.2

Table of contents

## Test report - version 0.2

### 1 Introduction

#### 1.1 Purpose of application

#### 1.2 General characteristics of application

### 2 Test environment

#### 2.1 Hardware environment

#### 2.2 Software environment and system information

##### 2.2.3 Software settings

### 3 Known bugs and limitations

Bug issue #11 (If the mobile lacks internet connection it force closes) is known. If the mobile lacks internet connection, the application experiences an error and shuts down.

### 4 Test specifications for this test report

#### 4.1 Non-functional tests

### 5 Test report table

#### 5.1 User tests

### 6 UNIT-tests

*Application version: 0.2*

*Date: 2012-05-17*

*Authors: Oscar Brodefors, Filip Askviken, Rikard Andersson,  
Emil Nyström*

*This version overrides all previous versions.*

# 1 Introduction

## 1.1 Purpose of application

A simple application for smartphones where students can gather to sell and buy used books needed at courses given at Chalmers (and possibly other Universities). Users can register books for sale and other users can search and find these books. The unique selling point compared to existing sites like blocket or tradera is the niche market. When buying from our application you know that the seller is close by and that you can set up a meeting to get the book the same day.

## 1.2 General characteristics of application

No transactions are handled by the application. The application simply matches buyer with seller and provides the opportunity for market mechanisms to function in the interaction between these two parties.

Functionality to add a plethora of attributes about the book and for users to search for books on these criterias. Also the ability to describe a book in general and to upload a photo of the book taken with the mobile camera. A book could be uploaded by filling in a ISBN-number and then obtaining book title, author and edition.

The ability for other users to comment on the book.

Possible functions to be added in future versions:

- Auctions
- Connection to Cremona to compare price and to buy new books
- A website connected to the application

## 2 Test environment

This android-project is preferably tested and runned from Eclipse. First step for a new tester or developer should be to install EGit from Eclipse Marketplace and the ADT plugin for Eclipse (<http://developer.android.com/sdk/eclipse-adt.html>). Please visit and read the installation guide (<http://developer.android.com/sdk/installing.html>).

After installation of Android ADT and connection with Github in your Eclipse, import the project in Eclipse: file → import

→ Git → Projects from Git → next → URL → insert <https://emil-nystrom@github.com/oscarbr/Brainiacs-startup.git> as URL → next → next → next → finish. Now you are ready to build or run the application. The application is runned through a .apk-package. The apk-package includes the release 0.1 which is the version tested in this test report.

The .apk-package can be found at <https://github.com/oscarbr/Brainiacs-startup>, go to tags → RELEASE\_0.1.zip.

The test report document to new developers and testers is available on: <https://docs.google.com/document/d/1oXC8qSHJzy3TRIEg0nVonFR0eNtPC01yBdP7oxgW0lc/edit>.

The test report table is available at: <https://docs.google.com/spreadsheets/cc?key=0Ao75Xu-pWZuDdGxoc1ZROF9BZX1EUThuUnJ5Uu11OHc>.

## 2.1 Hardware environment

For the test, three MacBook's and one Dell XPS M1530 have been used. Three computers with the operating software Mac OSX-Lion and one with Windows 8 have been used. A real mobile phone (Samsung Galaxy S plus) have been used for all the tests, except for the test NF3 where a Eclipse emulator (Target Android 2.1, skin WVGA800) have been used.

## 2.2 Software environment and system information

For the test, the following software have been used:

- Eclipse SDK version 3.7.2
- Android Development Toolkit version 18.0.0
- Eclipse Mylyn project version 3.6.5
- Eclipse EGit 1.3.0

### 2.2.3 Software settings

No specific settings needed, use default settings.

## 3 Known bugs and limitations

Bug issue #11 (If the mobile lacks internet connection it force closes) is known. If the mobile lacks internet connection, the application experiences an error and shuts down.

Bug issue #33 (NOMAD blocking Libris request) is known. If the phone is connected via NOMAD (or any network that require login via web browser).

Bug issue #36 (search short list sorts on ad-date instead of price) found via test 2.1.4-P.

## 4 Test specifications for this test report

### Test ID 0.1-P **Startable application**

Description: The application is startable on the phone.

Precondition: The tester have the application installed.

Test steps:

1. The tester presses the application in the phone menu
2. The tester verifies that the application starts without any exception messages.

Postcondition: The application is runnable

Related requirements: NF0

### Test ID 1.1-P **Book registration**

Description: A seller can register a book for sale

Precondition: Test ID 0.1. The seller have a book to sell, have the application and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text boxes:

Author: Daniel Defoe

Book title: Robinson Crusoe

ISBN:

Version: 2nd edition

Publishing year: 1999

Course: literature

Price: 300 SEK

Comments: excellent quality

3. The tester presses "add" and the book is added to the market.

Postcondition: A book is uploaded to the market.

Related requirements: 1.1

### Test ID 1.1-N **Book registration without required information**

Description: A seller can register a book for sale  
Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text boxes:

Author: Daniel Defoe

Book title:

ISBN:

Version: 2nd edition

Publishing year: 1999

Course: literature

Price:

Comments: excellent quality

3. The tester presses "add".

Postcondition: The tester gets a error message saying "missing title and price".

Related requirements: 1.1

#### Test ID 1.1.1.1-P **Pricing a book - 99 SEK**

Description: The seller can set a price for the book

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text box "price": 99 SEK
3. The tester presses "add".

Postcondition: A book is uploaded to the market with the correct price.

Related requirements: 1.1.1

#### Test ID 1.1.1.2-P **Pricing a book - 300 SEK**

Description: The seller can set a price for the book

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text box "price": 300 SEK
3. The tester presses "add".

Postcondition: A book is uploaded to the market with the correct price.

Related requirements: 1.1.1

Test ID 1.1.1.3-P **Pricing a book - 1000 SEK**

Description: The seller can set a price for the book

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text box "price": 1000 SEK
3. The tester presses "add".

Postcondition: A book is uploaded to the market with the correct price.

Related requirements: 1.1.1

Test ID 1.1.1.4-P **Pricing a book - 0 SEK**

Description: The seller can set a price for the book

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text box "price": 0 SEK
3. The tester presses "add".

Postcondition: A book is uploaded to the market with the correct price.

Related requirements: 1.1.1

Test ID 1.1.1.1-N **Pricing a book - hundra kronor**

Description: The seller can set a price for the book

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the textbox "price": hundra kronor

3. The tester presses "add".

Postcondition: The tester should get a error message stating: "Invalid price, please write a number between 0-10000 SEK.

Related requirements: 1.1.1

Test ID 1.1.1.2-N **Pricing a book - 133700 SEK**

Description: The seller can set a price for the book

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the textbox "price": 133700 SEK
3. The tester presses "add".

Postcondition: The tester should get a error message stating: "Invalid price, please write a number between 0-10000 SEK.

Related requirements: 1.1.1

Test ID 1.1.1.3-N **Pricing a book - no input**

Description: The seller can set a price for the book

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the textbox "price": no input
3. The tester presses "add".

Postcondition: The tester should get a error message stating: "Invalid price, please write a number between 0-10000 SEK.

Related requirements: 1.1.1

Test ID 1.1.2-P **Picture uploading**

Description: The seller can upload a picture of the book for sale

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details, and wants to show a picture of the book.

Test steps:

1. The tester presses the button "add book"
2. The tester fills in the information details:  
Author: Daniel Defoe  
Book title: Robinson Crusoe  
ISBN:  
Version: 2nd edition  
Publishing year: 1999  
Course: literature  
Price: 300 SEK  
Comments: excellent quality

3. The user then presses the button "Photo" and takes a photo of a book.
4. The tester presses the button "add" and the book is added to the market.

Postcondition: A book in the market has a picture connected to it.  
Related requirements: 1.1.2

Test ID 1.1.3-P **Title is required, >= four characters**

Description: The seller has to set a title for the book  
Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.  
Test steps:  

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text boxes:  
Author: Daniel Defoe  
Book title: Robinson Crusoe  
Price: 300

3. The tester presses "add".

Postcondition: Every added book have a title.  
Related requirements: 1.1.3

Test ID 1.1.3-N **Title is required, < four characters**

Description: The seller has to set a title for the book  
Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.  
Test steps:  

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text boxes:  
Author: Daniel Defoe



Book title: xxyy  
Price: 300

3. The tester presses "add".

Postcondition: A book cannot be added without a title and the tester gets a exception message "Title must be greater than 10 characters".

Related requirements: 1.1.3

#### Test ID 1.1.4-P **ISBN number**

Description: The seller can set an ISBN for the book for sale

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text boxes:

Author: Jan Skansholm

Book title: Java direkt med Swing

ISBN: 9789144060743

Version:

Publishing year:

Price: 300

3. The tester presses "add".

Postcondition: A added book have a ISBN number

Related requirements: 1.1.4

#### Test ID 1.1.4-N **ISBN number**

Description: The seller can set an ISBN for the book for sale

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text boxes:

Author: Jan Skansholm

Book title: Java direkt med Swing

ISBN: 123456

Version:

Price: 300

3. The tester presses "add".

Postcondition: The tester gets a exception message: "The ISBN-number must be 10 or 13 digits long."

Related requirements: 1.1.4

Test ID 1.1.5-P **Publishing year**

Description: The seller can set a publishing year for the book for sale.

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.

2. The tester then fills in the following information details in the text boxes:

Author: Daniel Defoe

Book title: testing the title

ISBN:

Version: 2nd edition

Publishing year: 1999

Course: literature

Price: 300

3. The tester presses "add".

Postcondition: A added book have a publishing year

Related requirements: 1.1.5

Test ID 1.1.5-N **Publishing year**

Description: The seller can set a publishing year for the book for sale.

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.

2. The tester then fills in the following information details in the text boxes:

Author: Daniel Defoe

Book title: testing the title

ISBN:

Version: 2nd edition

Publishing year: 2015

Course: literature

Price: 300

3. The tester presses "add".

Postcondition: The tester gets a exception message: "this is not a valid publishing year".

Related requirements: 1.1.5

Test ID 1.1.6.1-P **Book connected to a course - course name**

Description: The seller can set a course for which he/she used the book.

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

The tester presses the button for adding a new book.

The tester then fills in the following information details in the text boxes:

Author: Emil Nyström

Book title: Databases rule

Course: Databases

Price: 300

3. The tester presses "add".

Postcondition: A book is added with a course connected to it.

Related requirements: 1.1.6

Test ID 1.1.6.2-P **Book connected to a course - course code**

Description: The seller can set a course for which he/she used the book.

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.

2. The tester then fills in the following information details in the text boxes:

Author: Emil Nyström

Book title: Databases rule

Course: TDA357

Price: 300

3. The tester presses "add".

Postcondition: A book is added with a course connected to it.

Related requirements: 1.1.6

Test ID 1.1.7-P **Comments**

Description: The seller can add a comment about the book.

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text boxes:

Author: Emil Nyström

Book title: Databases rule

Price: 300

Comments: excellent quality

3. The tester presses "add".

Postcondition: A book is added with a comment connected to it.

Related requirements: 1.1.7

Test ID 1.1.8-N **Author**

Description: The seller can add a author to the book.

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text boxes:

Author: My

Book title: Databases rule

Price: 300

Comments: excellent quality

3. The tester presses "add".

Postcondition: The tester should get a exception message saying "a author must be five characters or more"

Related requirements: 1.1.8

Test ID 1.2-P **Taking a book off market**

Description: The seller can take a book off the market.

Precondition: Test ID 0.1, 1.1. The tester have a book advertisement.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text boxes:

Author: Daniel Defoe  
Book title: Robinson Crusoe  
Edition:  
Publishing year:  
Course: literature  
Price: 300  
Comments: excellent quality

3. The tester presses "add".  
The book should then be uploaded to the market.  
4. The tester tries to delete the ad by pressing the edit-button at the market and use his/hers password. In the ad-view, the user presses "delete" and deletes the ad.

Postcondition: The book should be deleted from the market.  
Related requirements: 1.2

Test ID 1.2-N **Taking a book off market**

Description: The seller can take a book off the market.  
Precondition: Test ID 0.1, 1.1. The tester have a book advertisement.

Test steps:

1. The tester presses the button for adding a new book.  
2. The tester then fills in the following information details in the text boxes:

Author: Daniel Defoe  
Book title: Robinson Crusoe  
Edition:  
Publishing year:  
Course: literature  
Price: 300  
Comments: excellent quality

3. The tester presses "add".  
The book should then be uploaded to the market.  
4. The tester tries to delete the ad by pressing the edit-button at the market and use his/hers password. The tester fills in a wrong password.

Postcondition: The book is not editable and should be deleted from the market within two months.  
Related requirements: 1.2

Test ID 1.3-P **Updating information about a book**

Description: The seller can update information about a book.

Precondition: Test ID 0.1, 1.1. The tester have a book advertisement.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text boxes:

Author: Daniel Defoe

Book title: Robinson Crusoe

Edition:

Publishing year:

Course: literature

Price: 300

Comments: excellent quality

3. The tester presses "add".

The book should then be uploaded to the market.

4. The tester then tries to edit the ad by pressing the edit-button at the market and use his/hers password. The tester changes the author to Astrid Lindgren and then presses the button "save".

Postcondition: The book is editable and the author is changed to Astrid Lindgren

Related requirements: 1.2

#### Test ID 2.1.1-P **Search by title**

Description: In a search, a buyer can specify a book's title.

Precondition: Test ID 0.1. The tester are looking for a specific book at the market and the book "Robinson Crusoe" by Daniel Defoe is added to the market.

Test steps:

1. The tester presses the button for searching a book.
2. The tester then fills in the following information details in the text boxes:

Author:

Book title: Robinson Crusoe

Course:

3. The tester presses the button "search".

Postcondition: The tester should get a match and find the book.

Related requirements: 2.1.1

#### Test ID 2.1.1-N **Search by title**

Description: In a search, a buyer can specify a book's title.

Precondition: Test ID 0.1. The tester are looking for a

specific book at the market

Test steps:

1. The tester presses the button for searching a book.
2. The tester then fills in the following information details in the text boxes:

Author:

Book title: Petter och hans fyra getter

Course:

3. The tester presses the button "search".

Postcondition: The tester will not get a match in the market with the following message: "your search didn't match any books at the market".

Related requirements: 2.1.1

#### Test ID 2.1.2-P **Search by author**

Description: In a search, a buyer can specify a book's author.

Precondition: Test ID 0.1. The tester are looking for a specific book at the market. The Book "Robinson Crusoe" with the author "Daniel Defoe" is in the market.

Test steps:

1. The tester presses the button for searching a book.
2. The tester then fills in the following information details in the text boxes:

Author: Daniel Defoe

Book title:

Course:

3. The tester presses the button "search".

Postcondition: The tester gets a match and finds the book "Robinson Crusoe"

Related requirements: 2.1.2

#### Test ID 2.1.2-N **Search by author**

Description: In a search, a buyer can specify a book's author.

Precondition: Test ID 0.1. The tester are looking for a specific book at the market

Test steps:

1. The tester presses the button for searching a book.
2. The tester then fills in the following information details in the text boxes:

Author: Emil Nyström

Book title:

Course:

3. The tester presses the button "search".

Postcondition: The tester will not get a match in the market, and get the following exception message: "your search didn't match any books at the market".

Related requirements: 2.1.2

Test ID 2.1.3.1-P **Search by course - course name**

Description: In a search, a buyer can specify which course the book should be used in

Precondition: Test ID 0.1. The tester are looking for a specific book at the market. The book "Java direkt med Swing" for the course "objektorienterad programmering" is in the market. The book "Linear algebra and its applications" for the course TMV215 is in the market.

Test steps:

1. The tester presses the button for searching a book.
2. The tester then fills in the following information details in the text boxes:

Author:

Book title:

Course: objektorienterad programmering

3. The tester presses the button "search".

Postcondition: The tester will get a match on the search: "Java direkt med Swing".

Related requirements: 2.1.3

Test ID 2.1.3.2-P **Search by course - course code**

Description: In a search, a buyer can specify which course the book should be used in

Precondition: Test ID 0.1. The tester are looking for a specific book at the market. The book "Java direkt med Swing" for the course "objektorienterad programmering" is in the market. The book "Linear algebra and its applications" for the course TMV215 is in the market.

Test steps:

1. The tester presses the button for searching a book.
2. The tester then fills in the following information details in the text boxes:

Author:

Book title:



Course: TMV215

3. The tester presses the button "search".

Postcondition: The tester will get a match on the search:  
"Linear algebra and its applications".

Related requirements: 2.1.3

#### Test ID 2.1.3-N **Search by course**

Description: In a search, a buyer can specify which course the book should be used in

Precondition: Test ID 0.1. The tester are looking for a specific book at the market. The book "Java direkt med Swing" for the course "objektorienterad programmering" is in the market. The book "Linear algebra and its applications" for the course TMV215 is in the market.

Test steps:

1. The tester presses the button for searching a book.
2. The tester then fills in the following information details in the text boxes:

Author:

Book title:

Course: Harry Potter Knowledge

3. The tester presses the button "search".

Postcondition: The tester will not get a match in the market with the following message: "your search didn't match any books at the market".

Related requirements: 2.1.3

#### Test ID 2.1.4-P **Search sorting**

Description: A search can be sorted on any of the available attributes that a book can have. Originally it is sorted by price.

Precondition: Test ID 0.1. The tester are looking for a specific book at the market. Three ads of the book "Robinson Crusoe" is in the market with the prices: 300, 100, 500.

Test steps:

1. The tester presses the button for searching a book.
2. The tester then fills in the following information details in the text boxes:

Author:

Book title: Robinson Crusoe

Course:

3. The tester presses the button "search".

Postcondition: The tester gets three books as result with the cheapest first (100, 300, 500).

Related requirements: 2.1.4

#### Test ID 2.2-P **Buying requests**

Description: A buyer can upload a buy request

Precondition: Test ID 0.1, installed application.

Test steps:

1. The tester opens the application
2. The tester presses the button for "add a request"
3. The tester fills in the information regarding the request:

Author: Daniel Defoe

Book title: Robinson Crusoe

ISBN:

Version: 2nd edition

Publishing year: 1999

Course: literature

Price: 300 SEK

Comments: excellent quality

4. The tester presses the button "add request"

Postcondition: A buying request is added to the market.

Related requirements: 2.2

#### Test ID 3.1-P **Book added by ISBN-number**

Description: A book can be uploaded by ISBN-number.

Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.

Test steps:

1. The tester presses the button for adding a new book.
2. The tester then fills in the following information details in the text boxes:

Author:

Book title:

ISBN: 9789144060743

Version:

Publishing year:

Course: Objektorienterad programmering

Price: 300 SEK

Comments:

3. The tester presses "add".

Postcondition: The tester gets the title: "Java direkt med Swing" and the author "Jan Skansholm".  
Related requirements: 3.1, 3.2

Test ID 3.1-N **Book added by ISBN-number**

Description: A book can be uploaded by ISBN-number.  
Precondition: Test ID 0.1. The tester have a book to sell, have the application, and have filled in user details.  
Test steps:  
1. The tester presses the button for adding a new book.  
2. The tester then fills in the following information details in the text boxes:  
Author:  
Book title:  
ISBN: 1111111111  
Version:  
Publishing year:  
Course: Objektorienterad programmering  
Price: 300 SEK  
Comments:

3. The tester presses "add".

Postcondition: The tester gets a error message in the text box for author: "No title found".  
Related requirements: 3.1, 3.2

Test ID 6.1-P **Book sold by auction**

Description: A seller can sell his/hers book on auction  
Precondition: Test ID 0.1, 1.1  
Test steps:  
1. The tester presses the button for adding a new book.  
2. The tester presses the button "sell in auction"  
3. The tester then fills in the following information details in the text boxes:  
Author: Daniel Defoe  
Book title:  
ISBN:  
Version: 2nd edition  
Publishing year: 1999  
Course: literature  
Price:  
Comments: excellent quality

3. The tester presses "fill in auction-details".  
4. The tester fills in auction details: startdate: 2012-06-

01, enddate: 2012-06-15 and start amount: 50 SEK.

5. Tester presses "add book"

Postcondition: A book is uploaded at the market and sold by auction.

Related requirements: 6.1

Test ID 6.1-N **Auction, invalid date**

Description: A seller can sell his/hers book on auction

Precondition: Test ID 0.1, 1.1

Test steps:

1. The tester presses the button for adding a new book.

2. The tester presses the button "sell in auction"

3. The tester then fills in the following information details in the text boxes:

Author: Daniel Defoe

Book title:

ISBN:

Version: 2nd edition

Publishing year: 1999

Course: literature

Price:

Comments: excellent quality

3. The tester presses "fill in auction-details".

4. The tester fills in auction details: startdate: 2012-01-01, enddate: 2012-06-15, start amount: 50 SEK, reservation price:

5. Tester presses "add book"

Postcondition: The tester gets a exception

message: "Invalid date, a auction must start after the present date".

Related requirements: 6.1

Test ID 6.2-P **Book sold by auction - reservation price**

Description: A seller can sell his/hers book on auction

Precondition: Test ID 0.1, 1.1

Test steps:

1. The tester presses the button for adding a new book.

2. The tester presses the button "sell in auction"

3. The tester then fills in the following information details in the text boxes:

Author: Daniel Defoe

Book title:

ISBN:

Version: 2nd edition

Publishing year: 1999  
Course: literature  
Price:  
Comments: excellent quality

3. The tester presses "fill in auction-details".
4. The tester fills in auction details: startdate: 2012-06-01, enddate: 2012-06-15, start amount: 50 SEK, reservation price: 300 SEK
5. Tester presses "add book"

Postcondition: A book is uploaded at the market and sold by auction with a reservation price.  
Related requirements: 6.2

## 4.1 Non-functional tests

### Test ID NF1 **Usability**

Description: The tester is checking the usability of the application.

Test steps: The tester should ask someone who hasn't used the application before and ask them to add a book. and take time of the process and ask questions afterwards, in order to understand if the application is considered easy to understand. Read more under headline 7 regarding black box testing.

Postcondition: The application is easy to understand and use.  
Related requirements: NF1

### Test ID NF2 **Usability**

Description: The tester is checking the usability of the application.

Test steps: The tester identifies one person who does not understand a part of the application, and then asks the person to read in the help-view. The person does this and then understands how to use the application. The tester also reads the information in the help-view and make sure it's understandable and correct.

Postcondition: The application is easy to understand and use.  
Related requirements: NF2

### Test ID NF3 **Reliability**

Description: The tester is checking the reliability of the application.

Test steps: The tester will add books at the same time as three other testers. This small test will indicate if the server can handle four uploading books at the same time. The testers will use the android emulator in Eclipse for these tests.

Postcondition: The application is reliable.

Related requirements: NF3

#### Test ID NF4 **Performance**

Description: The tester is checking the performance of the application.

Test steps: The tester should view information about a book and then turn off the internet connection the the phone. The tester should then open the application and still see the information on the book viewed last time.

Postcondition: The application har good performance.

Related requirements: NF4

#### Test ID NF6 **Implementation**

Description: The tester is checking the implementation and size of the application.

Test steps: After every release, the tester should control that the client-side application cannot exceed 10 MB.

Postcondition: The application is slim.

Related requirements: NF6

## 5 Test report table

The tested application is in ... länk

### 5.1 User tests

Date	Test ID	Result	Comments
120517	0.1	Passed	without remarks
120517	1.1-P	Passed	without remarks
120517	1.1-N	Passed	without remarks
120517	1.1.1.1-P	Passed	without remarks
120517	1.1.1.2-P	Passed	without remarks
120517	1.1.1.3-P	Passed	without remarks
120517	1.1.1.4-P	Passed	without remarks
120517	1.1.1.1-N	Passed	"hundra kronor" cannot be filled in.
120517	1.1.1.2-N	Passed	without remarks
120517	1.1.1.3-N	Passed	without remarks
120517	1.1.2-P	No result	not implemented
120517	1.1.3-P	Passed	without remarks
120517	1.1.3-N	Passed	without remarks
120517	1.1.4-P	Passed	without remarks
120517	1.1.4-N	Passed	without remarks

120517	1.1.5-P	Passed	without remarks
120517	1.1.5-N	Passed	Without remarks
120517	1.1.6.1-P	Passed	without remarks
120517	1.1.6.2-P	Passed	without remarks
120517	1.1.7-P	Passed	without remarks
120517	1.1.8-N	Passed	without remarks
120517	1.2-P	No result	not implemented
120517	1.2-N	No result	not implemented
120517	1.3-P	No result	not implemented
120517	2.1.1-P	Passed	without remarks
120517	2.1.1-N	Passed	without remarks
120517	2.1.2-P	Passed	without remarks
120517	2.1.2-N	Passed	without remarks
120517	2.1.3.1-P	Passed	without remarks
120517	2.1.3.2-P	Passed	without remarks
120517	2.1.3-N	Passed	without remarks
120517	2.1.4-P	Failed	the app sorts by ad-date (-> issue #36)
120517	2.2-P	No result	not implemented
120517	3.1-P	Passed	without remarks
120517	3.1-N	Passed	without remarks
120517	6.1-P	No result	not implemented
120517	6.1-N	No result	not implemented
120517	6.2-P	No result	not implemented
120517	NF1	Passed	without remarks, app easy to use.
feedback			led to change in RID: 1.1.8 - >= 5 characters for a author
120517	NF2	No result	not implemented
120517	NF3	Passed	without remarks
120517	NF4	Passed	without remarks
120517	NF6	Passed	without remarks

## 6 UNIT-tests

### Coverage Summary

name	class	method	block	line
itBrainiacs.muffins	61,5% <span>8/13</span>	60,2% <span>50/83</span>	42,3% <span>982/2323</span>	41,7% <span>263/630</span>

### Coverage Breakdown by Source File

name	class	method	block	line
<a href="#">AddBookActivity.java</a>	100,0% <span>2/2</span>	60,0% <span>6/10</span>	57,8% <span>237/410</span>	58,3% <span>51/88</span>
<a href="#">DataBook.java</a>	100,0% <span>2/2</span>	79,5% <span>35/44</span>	50,7% <span>181/357</span>	54,6% <span>65/119</span>
<a href="#">DataBookFactory.java</a>	100,0% <span>1/1</span>	80,0% <span>4/5</span>	68,7% <span>412/600</span>	54,1% <span>112/207</span>
<a href="#">MuffinsActivity.java</a>	0,0% <span>0/1</span>	0,0% <span>0/2</span>	0,0% <span>0/82</span>	0,0% <span>0/16</span>
<a href="#">MuffinsUtility.java</a>	100,0% <span>1/1</span>	50,0% <span>1/2</span>	58,7% <span>37/63</span>	42,7% <span>6/15</span>
<a href="#">SearchActivity.java</a>	100,0% <span>1/1</span>	50,0% <span>2/4</span>	45,5% <span>50/110</span>	40,7% <span>11/27</span>
<a href="#">SearchItemActivity.java</a>	0,0% <span>0/1</span>	0,0% <span>0/3</span>	0,0% <span>0/158</span>	0,0% <span>0/34</span>
<a href="#">SearchResultActivity.java</a>	0,0% <span>0/2</span>	0,0% <span>0/4</span>	0,0% <span>0/129</span>	0,0% <span>0/26</span>
<a href="#">ServerCommunicator.java</a>	100,0% <span>1/1</span>	50,0% <span>2/4</span>	73,0% <span>65/89</span>	60,0% <span>17/28</span>
<a href="#">SettingsActivity.java</a>	0,0% <span>0/1</span>	0,0% <span>0/5</span>	0,0% <span>0/325</span>	0,0% <span>0/70</span>

The test-project can be found in the folder Muffins/tests

Extraction from Jenkins build and run of tests:

```
[echo] Running tests ...
[exec] itBrainiacs.muffins.test.ActivityTests.AddBookActivityTest:.
[exec] itBrainiacs.muffins.test.ActivityTests.SearchActivityTest:.
[exec] itBrainiacs.muffins.test.DataBookFactoryTest:...
[exec] Failure in testParseLibris:
[exec] junit.framework.ComparisonFailure: expected:<Sagan om ringen> but was:<"Sagan om ringen">
[exec] at itBrainiacs.muffins.test.DataBookFactoryTest.testParseLibris(DataBookFactoryTest.java:141)
[exec] at java.lang.reflect.Method.invokeNative(Native Method)
[exec] at android.test.AndroidTestRunner.runTest(AndroidTestRunner.java:169)
[exec] at android.test.AndroidTestRunner.runTest(AndroidTestRunner.java:154)
[exec] at android.test.InstrumentationTestRunner.onStart(InstrumentationTestRunner.java:430)
[exec] at android.app.Instrumentation$InstrumentationThread.run(Instrumentation.java:1447)
[exec] Failure in testSearchLibris:
[exec] junit.framework.ComparisonFailure: expected:<Sagan om ringen> but was:<"Sagan om ringen">
[exec] at itBrainiacs.muffins.test.DataBookFactoryTest.testSearchLibris(DataBookFactoryTest.java:110)
[exec] at java.lang.reflect.Method.invokeNative(Native Method)
[exec] at android.test.AndroidTestRunner.runTest(AndroidTestRunner.java:169)
[exec] at android.test.AndroidTestRunner.runTest(AndroidTestRunner.java:154)
[exec] at android.test.InstrumentationTestRunner.onStart(InstrumentationTestRunner.java:430)
[exec] at android.app.Instrumentation$InstrumentationThread.run(Instrumentation.java:1447)
[exec] .
[exec] itBrainiacs.muffins.test.DataBookTest:
[exec] Failure in testEquals:
[exec] junit.framework.AssertionFailedError: expected:<itBrainiacs.muffins.DataBook@44ece730> but
was:<itBrainiacs.muffins.DataBook@44ece788>
[exec] at itBrainiacs.muffins.test.DataBookTest.testEquals(DataBookTest.java:46)
[exec] at java.lang.reflect.Method.invokeNative(Native Method)
[exec] at android.test.AndroidTestRunner.runTest(AndroidTestRunner.java:169)
[exec] at android.test.AndroidTestRunner.runTest(AndroidTestRunner.java:154)
[exec] at android.test.InstrumentationTestRunner.onStart(InstrumentationTestRunner.java:430)
[exec] at android.app.Instrumentation$InstrumentationThread.run(Instrumentation.java:1447)
[exec] .
[exec] itBrainiacs.muffins.test.MuffinsUtilityTest:...
[exec] itBrainiacs.muffins.test.ServerCommunicatorTest:
[exec] Failure in testAddBook:
[exec] junit.framework.AssertionFailedError: Not yet implemented
[exec] at itBrainiacs.muffins.test.ServerCommunicatorTest.testAddBook(ServerCommunicatorTest.java:34)
[exec] at java.lang.reflect.Method.invokeNative(Native Method)
[exec] at android.test.AndroidTestRunner.runTest(AndroidTestRunner.java:169)
[exec] at android.test.AndroidTestRunner.runTest(AndroidTestRunner.java:154)
[exec] at android.test.InstrumentationTestRunner.onStart(InstrumentationTestRunner.java:430)
[exec] at android.app.Instrumentation$InstrumentationThread.run(Instrumentation.java:1447)
```

## 7 Automated build

Automated tests via Jenkins was successful:

**BUILD SUCCESSFUL**

Total time: 21 seconds