

# Social Document Library

Valerio Vaudi n.m. 1034003

# The Idea

- Starting to the evidence that in the IT World in almost every project somethings of new apperare. The idea was of build a Software that helps the IT Professionalist to understand how choice the best education matirials, technical book in particualr.
- Built a document library for IT Book
- Help the user in the search
- Help the user in book's evaluation though feedback and score of other user
- Built a employee application for upload book
- Built a employee application for manage book
  - update metadata, publish/unpublish, delete book

# User Target

- Analyze the results of the survey, the landscape of the usual professional-figure in the work world.
- Final User User-Profile
  - User profile
    - Age: 18 - 50
    - Education: wide kind of education
    - Work positions: wide kind of work positions
    - Years of experience: 0 - 30
    - Background: see the documentation
- Employee User User-Profile
  - User profile
    - Age: 18 – 50
    - Education: wide kind of education
    - Years of experience: 0 – 30
    - Background: see the documentation

# User Requirements Validation

- What
  - The idea
  - Core functions
  - User target
- How
  - Survey (sent by mail)

# Final User (1/2)

## Personas

- Junior user

*Junior Developer User*



**Name:** John Hilbert  
**Age:** 25 years old  
**Job Title:** Junior Java developer  
**Experience level:** less than 2 years  
**Work hours:** 9:00 am – 6:00 pm  
**Educations:** Bachelor degree  
**Technology:** Java, Sql, JavaScript  
**Family:** single

**Background:** John has started to learn a new technology, he has to learn Scala and the functional programming paradigms. John has a strong Java knowledge but don't has any details about the functional programming paradigms and is confused.

- Expert user

*Expert Developer User*



**Name:** Nicolas Freeman  
**Age:** 40 years old  
**Job Title:** Senior Developer  
**Experience level:** greater than 8 years  
**Work hours:** 9:00 am – 6:00 pm  
**Educations:** Master's degree in Computer Science  
**Technology:** Word, Excel, Project, SONAR, Jenkins, Visual Paradigms  
**Family:** Married

**Background:** Nicolas is a Software Architect that design solution for complex enterprise. In his work have be always updated. For this buy many books on Amazon of any category strictly technical, more general and so on, with his knowledge that he has accumulated over the years Nicolas can search in Amazon the books that her want buy. However sometimes, especially when don't has many experience in the a new scenario could be helpful a mean that could hint the right book for the job. For instance suppose that Nicolas work in a new project and have design an application rest base in which the front-end will be written in NodeJS based and the back-end will be in .NET. Nicolas would have a mean that can suggest the available books about how produce REST-WS in .NET and how consume REST-WS in NodeJS.

# Final User (2/2)

## Serenarios

### 1. ***A user should be able to search a book in the SDL***

John should start to learn the functional programming and Scala language, for a very important customer. Fortunately, John comes from the Java world, however he don't knows anything about Scala and more in general of Functional Programming. For this reason tries to search a book that can helps you in the learning process. John when return at home turn on your pc and log in to the document library. Once logged in John think on what search and click on the search function on menu. John now can start to search but is confused because don't know the name of a book about Scala and the worst problem is that don't know the name of the framework used in the Scala ecosystem. For repair on your leaks, John tries to use the search tag. For this search tries to choose Scala, Web, Functional Programming how search tag, then click on the search button, John in this way hopes that the search engine find for you books about Scala and all the search tags that he had choose. Wait some instant and under the search form compare the results of the search. Now John can assess the books reading the description, and the score of other users in order to built a his idea about which book could be a good choice for his intention.

### 2. ***A user should be able to read a book that is been searched in the SDL***

Nicolas is a senior developer and in his career studied very hard for stay up to date. Lately Nicholas must work on a new and complex project, in which many technologies must interact. In particular, the back-end implemented in .NET and the front-end will implemented in NodeJS. Since Nicholas don't know Node , he starts to search a good book for his study. Once that Nicholas had find the book that he consider good enough, he start to read the book page by page. Nicolas start the read when has change page something decide of use the keyboard command and other times, when he has the mouse in your hand use the button for forward the page that he find on the tool bar. After one hour. Nicolas see that is too late and he decide of turn-of the pc. Nicolas now for not forget the last page that he had read decide of put a bookmark with the specific command that is present on the tool bar.

### 3. ***A user should be able to tag the book in any time for perform a feedback on the book in term of rating and useful indicators for search***

Since Nicolas must always stay up to date due to his work, he decide of read a recent book on Scrum. He decide of take Essential Scrum. After some page, Nicolas is sure, Essential Scrum is a great book that explains very well all aspect and professionals figures in the Scrum framework. For this reason, he would like express his feedback. Nicolas access to the your my book list, search Essential Scrum in the list and click the button on the header of the feedback panel for open the feedback's dialog. When he click on the screen appear a popup with feedback title, feedback body and score. Nicolas typing on popup his impression and click on the ok button. In this way, any other search will include also the Nicolas feedback

# Employee User (1/2)

## Personas

### *Document Library Employee*



**Name:** Stacy Moore

**Age:** 30 years old

**Job Title:** Library Administrator

**Experience level:** greater than 3 years

**Work hours:** 9:00 am – 6:00 pm

**Educations:** Bachelor degree

**Technology:** Word, Excel, web

**Family:** single or married

**Background:** Stacy worked in the library for 5 years, his commitments are add and manages book's virtual shelves. She would like an

instrument for learn the trend of the library customer and add hints for the less popular books in order to understand if the book is useless for the library. In this way, stacy could understand the interests of customers and decide if add or not a new book in this area.

# Employee User (2/2)

## Serenarios

### **1. *An Employee user should be able to add or remove a book in the document library***

it's a normal Friday's afternoon, when at a certain point, Stacy get an e-mail in which the Stacy's chief asks of upload three new books in the library that were attached in the mail. Stacy then log-in in her company account, she click on book upload functionality. For upload a book Stacy has download the attachments list in her computer and then book by book, she uses the upload form. Book by book Stacy chooses the pdf master from his PC and reading the e-mail, type the author, description and tags in the search mask. When the process of book's processing will be complete, in the Stacy profile will appears a notification that info Stacy of the completion of process. From now Stacy could flag the book that wants to publish in the library. With the publish flag, in any times, Stacy may publish or unpublish the book.

### **2. *An Employee user should be able to update one or more book metadata in the document library***

It's a normal work day when Stacy for casually find a mistake in a metadata for Essential Scrum book. The book hasn't the correct search tags and if Stacy don't update this field probably an user couldn't be able to find out this book. For this reason Stacy open the option popup and modify the metadata adding Testing and Scrum how search tags then click on Ok button of popup. In this way, now the book will has also the correct metadata.

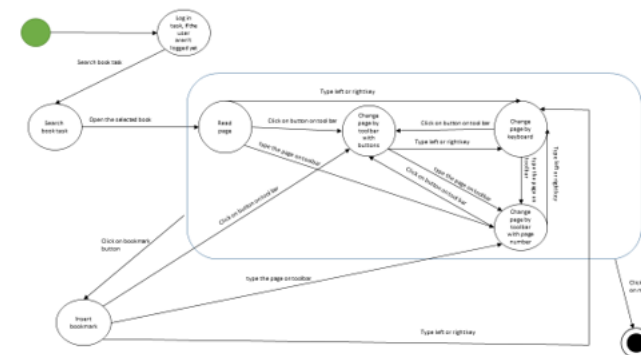
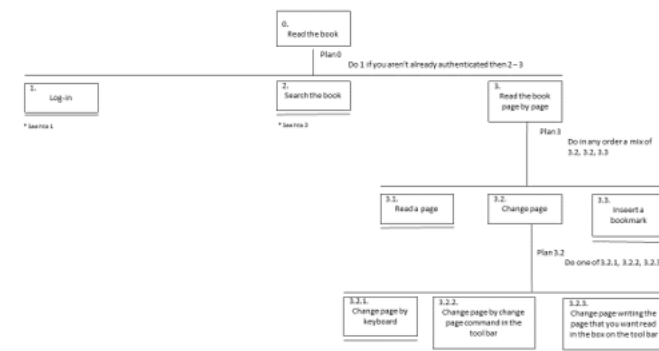
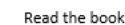


## Final-User

```

graph TD
    5[5. Add feedback and/or score] --> 1[1. Log in]
    5 --> 2[2. Choose the book]
    1 --> 1_note["* See Use 1"]
    2 --> 2_1[2.1. Use my book list]
    2 --> 2_2[2.2. Choose home on the menu]
    2 --> 2_3[2.3. Choose the book in the list]
    2_1 --> 2_1_1[2.1.1. Choose my book list on the menu]
    2_1 --> 2_1_2[2.1.2. Change the book name held on the list bar]
    2_3 --> 3_1[3.1. Open the create/update feedback module]
    2_3 --> 3_2[3.2. Add/update the score]
    2_3 --> 3_3[3.3. Add/update the feedback]
    2_3 --> 3_4[3.4. Confirm the update]
    3_1 --> 3_1_1[3.1.1. Add the feedback body]
    3_1 --> 3_1_2[3.1.2. Add the feedback title]
    3_2 --> 3_2_1["3.2.1. Do optionally in any order 3.2 and 3.3"]
    3_2 --> 3_2_2[3.2.2. Do 3.4]
    3_3 --> 3_3_1["3.3.1. Do optionally in any order 3.3.1 and 3.3.2"]
    3_3 --> 3_3_2[3.3.2. Confirm the update]
    3_4 --> 3_4_1["3.4.1. Do optionally in any order 3.3.1 and 3.3.2"]
    3_4 --> 3_4_2[3.4.2. Confirm the update]
  
```

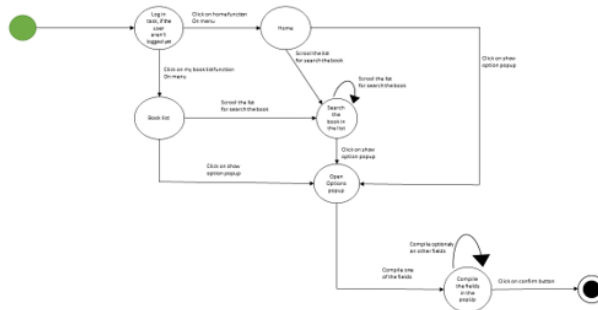
The diagram is a hierarchical Use Case Map (UCM) for a book review system. It starts with a root use case '5. Add feedback and/or score'. This use case has two main branches: '1. Log in' and '2. Choose the book'. The '1. Log in' use case has a note '\* See Use 1'. The '2. Choose the book' use case has three sub-use cases: '2.1. Use my book list', '2.2. Choose home on the menu', and '2.3. Choose the book in the list'. The '2.1. Use my book list' use case has two sub-use cases: '2.1.1. Choose my book list on the menu' and '2.1.2. Change the book name held on the list bar'. The '2.3. Choose the book in the list' use case has four sub-use cases: '3.1. Open the create/update feedback module', '3.2. Add/update the score', '3.3. Add/update the feedback', and '3.4. Confirm the update'. The '3.1. Open the create/update feedback module' use case has two sub-use cases: '3.1.1. Add the feedback body' and '3.1.2. Add the feedback title'. The '3.2. Add/update the score' use case has two sub-use cases: '3.2.1. Do optionally in any order 3.2 and 3.3' and '3.2.2. Do 3.4'. The '3.3. Add/update the feedback' use case has two sub-use cases: '3.3.1. Do optionally in any order 3.3.1 and 3.3.2' and '3.3.2. Confirm the update'. The '3.4. Confirm the update' use case has two sub-use cases: '3.4.1. Do optionally in any order 3.3.1 and 3.3.2' and '3.4.2. Confirm the update'. The diagram uses boxes for use cases, lines for relationships, and text for notes and plan numbers.



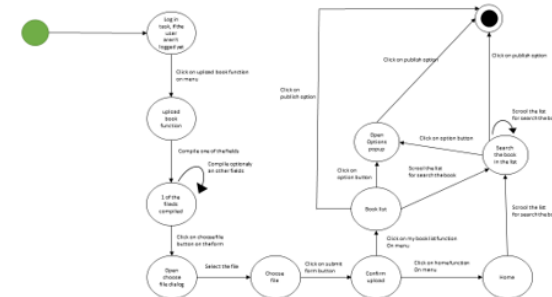
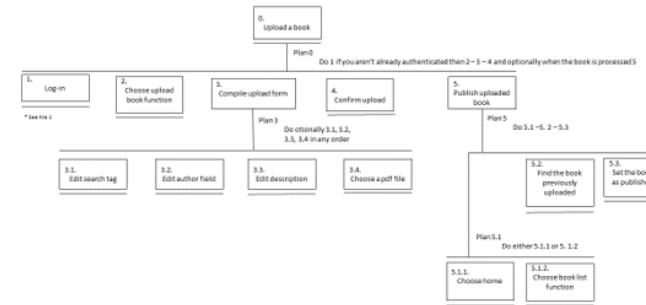
# HTA/STN (2/2)

## Employee-User

Manage book



Upload a book



# Prototype 1 (1/6)

## Expert evaluation

- Heuristic evaluation
  - Final-User
    - I must find a way for underline the functionality on the menu
      - recognition rather than recall: 3
    - I must reduce the size of the header for having more space and avoid more scroll
      - aesthetic and minimalistic design: 2
    - I must add the back functionality
      - flexibility of use: 3
    - Lacking help functionality
      - Help and documentation: 3
  - Employee-User
    - I must find a way for underline the functionality on the menu
      - recovery rather than recall: 3
    - Lacking help functionality
      - Help and documentation: 3
- Cognitive walkthrough
  - Final-User

The cognitive walkthrough has shown that the search functionality gateway is few visible
  - Employee-User

The cognitive walkthrough has shown that the upload book functionality gateway is few visible. Another issue that the technical evaluation has found is the possibility that a user may be confused as the file metadata when the file is essential and without it, the upload of the book should be in error.

# Prototype 1 (2/6)

## User evaluation

### Think Aloud

- Final-User user-profile
  - Search book
    - Popup for empty search
    - Read feedback section
  - Read book
    - Follower command tool bar
  - Add feedback/score
- Employee-User user-profile
  - Upload book
    - Notification system
    - Popup after successful upload

# Prototype 1 (3/6)

## Interview

- Final-User user-profile

- In the application, we have three functions: home, search and my book list. The home and my book list are very similar functions the first present the first 5 books and the second present all the book list adding a search function as facility. Could make sense have only the My book list function as home page and don't replicate the two functions?
- Could have sense transform the welcome header in some form of helper?
- Could have sense add a help button near the main things that aren't clear?
- Could be useful read the feedback of other users?
- Could be useful remove a book In the My preferred list section?

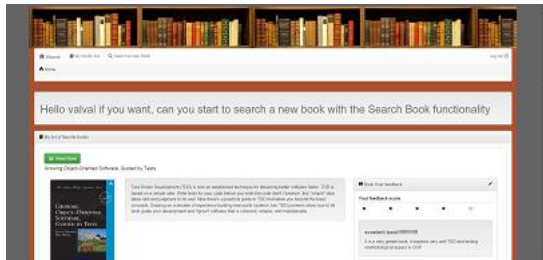
- Employee-User user-profile

- In the application, we have three functions: home, upload book and book list. The home and book list are very similar functions the first present the unpublished book and the second present all the book list. Could make sense have only the book list function as home page and don't replicate the two functions?
- In the book list functions does not exist the search function? Could have sense provide a search engine based on filter book that satisfy some criteria as name, is publish and so on?
- After the submit of an upload book job don't have an immediate feedback of the status of the job. Could have sense a notify function that info the employee of job's completes and status?

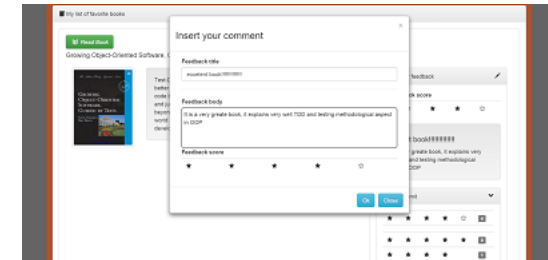
# Prototype 1 (4/6)

## Final-User User-Profile

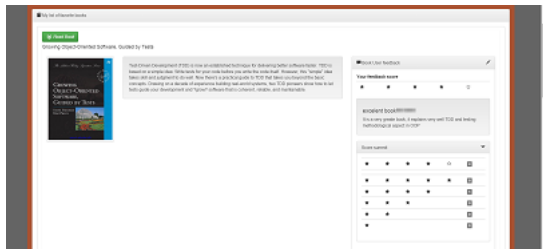
Home



Add/update  
Score/Feedback



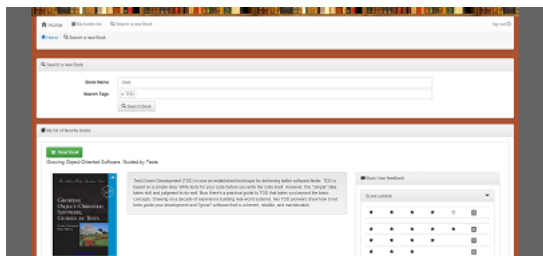
Book list



Read book



Book search



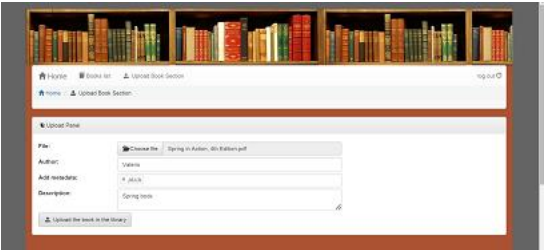
# Prototype 1 (5/6)

## Employee-User User-Profile

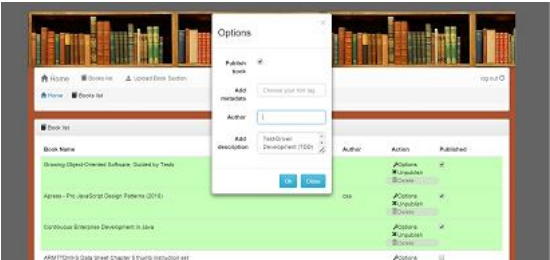
Home



Upload book



Manage book



# Prototype 1 (6/6)

## • **Todo-List of improvements for the second prototype**

### • **Final User interface**

- Transform the menu in a tool bar in which we have bigger icon for the functionality
- Introduce the Help functionality
- Replace the hello header with the Short help functionality
- Introduce the inline Help when necessary
- Introduce the Back functionality
- Introduce a feedback for the empty search
- Introduce the feedback reading functionality of other users
- Introduce the remove book functionality on the preferred list
- Remove the Home functionality with the first 5 books read, on the My book List functionality

### • **Employee interface**

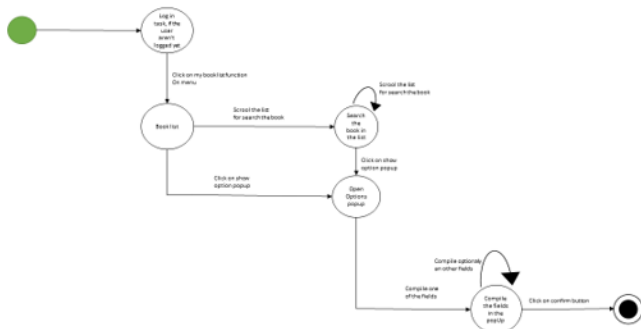
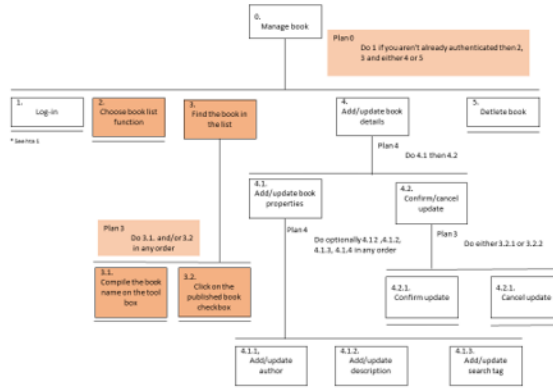
- Transform the menu in a tool bar in which we have bigger icon for the functionality
- Introduce the Help functionality
- Introduce Notifications functionality
  - Completion of the loading of the book
- Introduce a feedback for outcome of book's upload
- Remove the Home functionality with the unpublished books list
- Introduce a search form in the book list with the follow search criteria:
  - Book name
  - Book status: published, no published
- Introduce a means of underline in the upload form the presence of other metadata and a validation for prevents that a user submit the form without choosing the file
- Give separation between file and file metadata in the upload form, in order to give at user more emphasis to author, tags and description as metadata and file as a more meaningful field



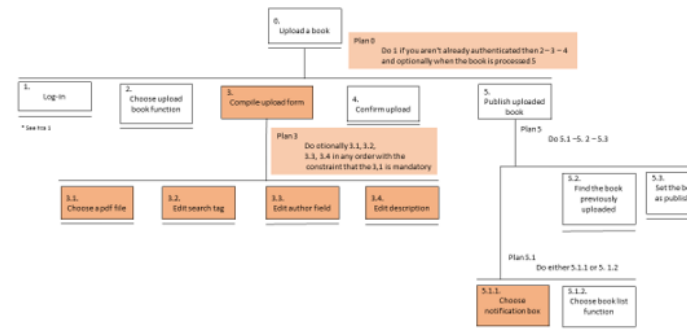


# HTA/STN Employee-User user-profile

Manage book

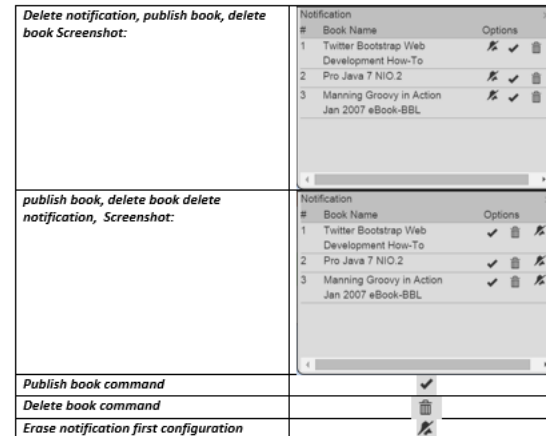


Upload book



# Prototype 2 (3/8)

- Controlled experiment
  - 2 configurations



- Hypothesis:

The configuration with delete notification, publish book, delete book how command position will make do less errors in publishing books on the fly in the notification panel, that delete notification, publish book, delete book how command position
- Null Hypothesis:

The number of errors in the publish books on the fly in the notification panel, in the two configurations does not changes
- Independent Variables (*controlled variable*)

Position of the erase notification command: two configurations how described above.
- Dependent Variables (*measured variable*)

Number of error due to the probable confusion that the erase notification command could raise in the two configuration.

# Prototype 2 (4/8)

- **Result of ANOVA analysis**

The result show that the hypothesis was correct, as proof we can read from the result that  $F$  is equals to 2.6 and  $F_{crit}$  is equals to 5.317655 and since that  $2.6 < 5.317655$  we can deduct that the publish book, delete book delete notification configuration is better than delete notification, publish book, delete book.

publish book, delete book delete notification	delete notification, publish book, delete book					
1	2					
3	6					
2	3					
1	1					
0	3					
Analisi varianza: ad un fattore						
RIEPILOGO						
Gruppi	Conteggio	Somma	Media	Varianza		
Colonna 1	5	7	1,4	1,3		
Colonna 2	5	15	3	3,5		
ANALISI VARIANZA						
Origine della variazione	SQ	gdl	MQ	F	Valore di significatività	F crit
Tra gruppi	6,4	1	6,4	2,666666667	0,141113281	5,317655
In gruppi	19,2	8	2,4			
Totale	25,6	9				

# Prototype 2 (5/8)

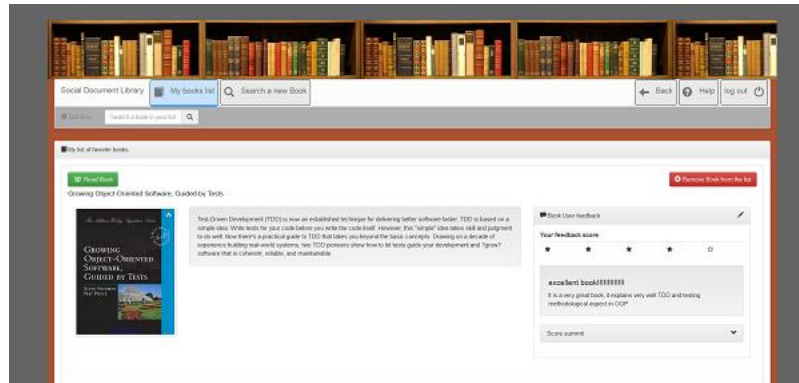
## **Post-task walkthroughs**

- Final-user user profile
  - Search a new book
  - Read a book
- Employee-user user profile
  - Upload book

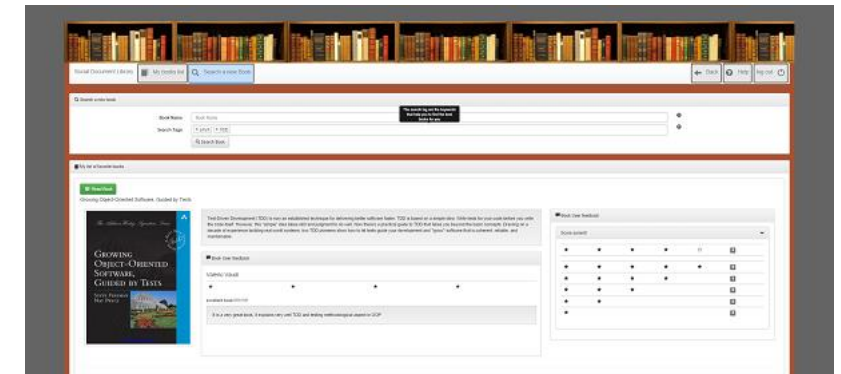
# Prototype 2 (6/8)

## Final-User User-Profile

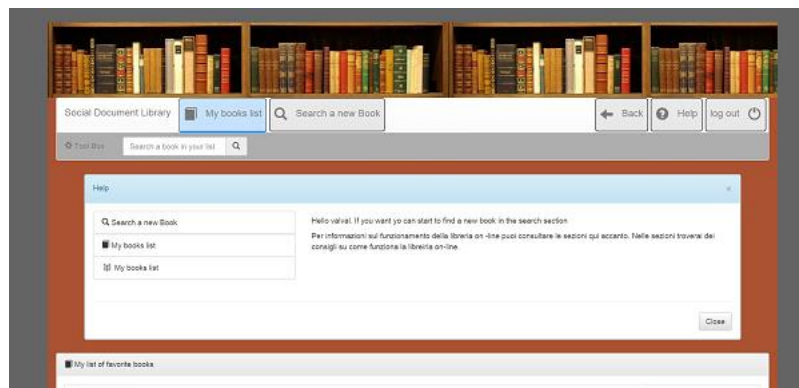
My Book List



Search a new Book



Help



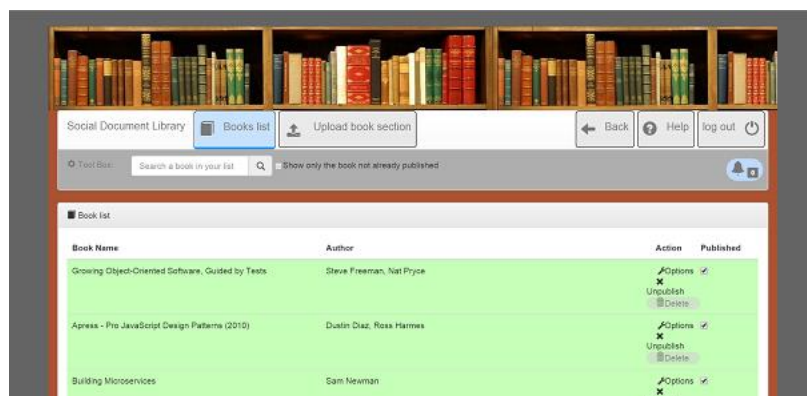
Read Book



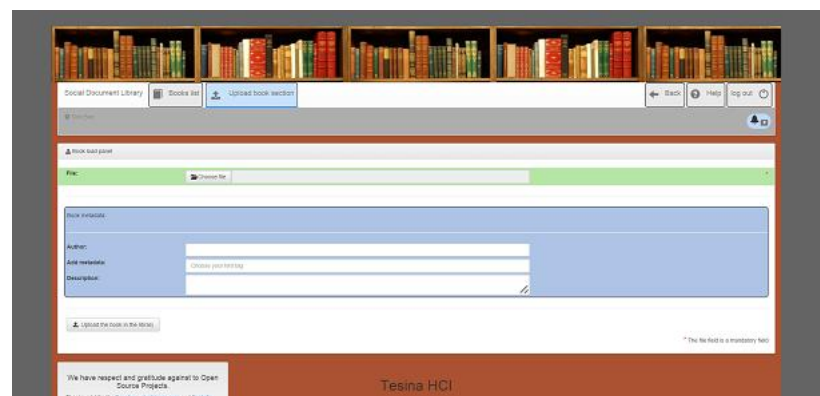
# Prototype 2 (7/8)

## Employee-User User-Profile

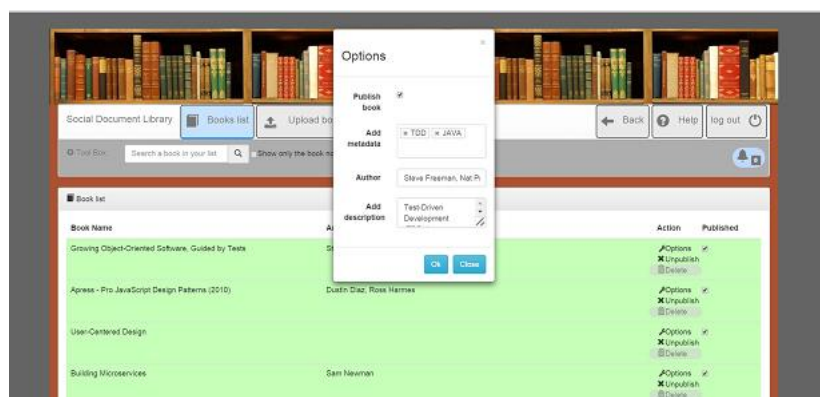
Book List



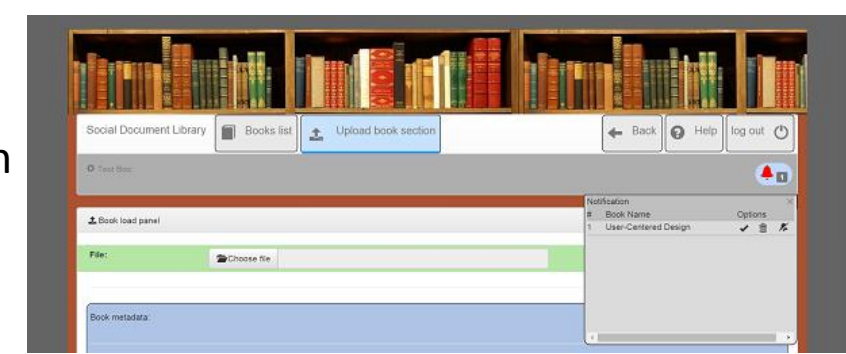
Upload a new Book



Managed Book



Notification System



# Prototype 2 (8/8)

## **Todo-List of improvements for the final product**

- **Final-user user profile**

- Move the find book in my book list function from the tool box to header panel of the my book list book.
- Add a change mouse cursor on the mouse hover on book reader command in the tool box and follower tool box

- **Employee-user user profile**

- Change the color of upload book button for make it more visible
- Add toll-tips on the publish book, delete book and delete notification on the notification container
- Add background color that appear on mouse hover on the publish book, delete book and delete notification on the notification container