
DOCUMENTO DI SPECIFICA DEI REQUISITI SOFTWARE

Comics Store Back-End

Enrico De Salve

Contents

1	Introduzione	3
1.1	Propositi	3
1.2	Obiettivi	3
1.3	Definizioni, acronimi ed abbreviazioni	4
1.4	Riferimenti	4
1.5	Panoramica	4
2	Descrizione generale	5
2.1	Prospettive del prodotto	5
2.2	Funzionalità del prodotto	5
2.3	Caratteristiche utente	6
2.4	Vincoli generali	6
3	Requisiti di interfaccia esterna	7
3.1	Interfaccia utente	7
3.2	Interfaccia hardware	7
3.3	Interfaccia software	7
3.4	Interfaccia di comunicazione	7
4	System Features	8
4.1	System Feature 1	8
4.1.1	Description and Priority	8
4.1.2	Stimulus/Response Sequences	8
4.1.3	Functional Requirements	8
4.2	System Feature 2 (and so on)	8
5	Other Nonfunctional Requirements	9
5.1	Performance Requirements	9
5.2	Safety Requirements	9
5.3	Security Requirements	9
5.4	Software Quality Attributes	9
5.5	Business Rules	10
6	Other Requirements	11
6.1	Appendix A: Glossary	11
6.2	Appendix B: Analysis Models	11
6.3	Appendix C: To Be Determined List	11

1 Introduzione

1.1 Propositi

Il proposito di questo documento è quello di specificare i requisiti del sistema software "Comics Store Back-End" per facilitarne la realizzazione e la validazione.

1.2 Obiettivi

È necessaria un'applicazione web per l'implementazione di servizi relativi alla gestione di un negozio online associato ad una fumetteria.

Bisogna fornire al gestore della fumetteria la possibilità di aggiungere (e rimuovere) collezioni di fumetti. Una collezione è caratterizzata da un nome, da un'immagine di copertina e dai fumetti che contiene. Il gestore deve poter aggiungere nuovi fumetti specificandone il numero dell'albo, il prezzo unitario, lo scrittore ed il disegnatore, il numero di pagine, il codice ISBN ed una descrizione. Per ogni fumetto si memorizza la quantità di copie disponibili nel catalogo; deve, naturalmente, essere possibile aggiungere copie di un fumetto incrementandone la quantità.

Ogni collezione fa parte di almeno una categoria, ma in generale ci possono essere più categorie associate ad una collezione. Una collezione è caratterizzata da un nome e da una breve descrizione. Il gestore deve poter creare ed eliminare le categorie ed associarle alle varie collezioni.

Il gestore della fumetteria deve poter, inoltre, creare nuovi sconti specificandone la percentuale di sconto e la data di scadenza e deve poter applicarli a vari fumetti.

Infine, il gestore deve poter richiedere la generazione di report indicanti, per un certo intervallo di tempo, le vendite.

Gli utenti (anche se non registrati) possono ricercare le collezioni per titolo, scrittore, disegnatore e codice ISBN; l'ordinamento delle collezioni ricercate deve poter essere deciso dall'utente. La ricerca per titolo deve mostrare anche le collezioni che corrispondono parzialmente all'input dell'utente.

Ad esempio se l'utente cerca la parola "attacco" l'applicazione deve mostrargli la collezione chiamata "L'attacco dei giganti" ma anche "Star Wars: l'attacco dei cloni" e tutte le collezioni che contengano nel nome la parola "attacco".

La ricerca mostra anche le collezioni contenenti fumetti che non sono disponibili per l'acquisto (sono terminate le copie).

Gli utenti possono registrarsi fornendo nome, cognome, data di nascita, indirizzo e-mail

e facoltativamente numero di telefono, paese e città in cui abitano.

Dal momento in cui si registra l'utente possiede un carrello.

Un utente registrato che ha effettuato l'accesso ha a disposizione tutte le altre funzionalità offerte dal sistema (oltre alla ricerca). Può aggiungere fumetti al suo carrello per poi acquistarli, naturalmente i fumetti devono essere disponibili al momento dell'acquisto. Il contenuto del carrello deve essere memorizzato così che quando l'utente effettua nuovamente l'accesso possa concludere l'acquisto, a seguito di un acquisto il carrello viene svuotato.

Un utente registrato può creare (e a seguito di ciò eliminare) delle liste dei desideri, sono liste nelle quali l'utente deve essere in grado di aggiungere qualunque fumetto (anche se non è attualmente disponibile nel catalogo). Per ogni lista l'utente può scegliere se vuole essere notificato sulle modifiche relative ai fumetti presenti nella lista desideri quali variazioni di prezzo, nuovi sconti o nuova disponibilità.

L'utente deve poter modificare alcune delle informazioni inserite in fase di registrazione (indirizzo e-mail, numero di telefono, paese e città).

1.3 Definizioni, acronimi ed abbreviazioni

BE Back-End

1.4 Riferimenti

Nel documento "Progettazione base di dati" è riportata la struttura di memorizzazione dei dati.

1.5 Panoramica

La restante parte di questo documento contiene una descrizione dettagliata delle funzionalità richieste al sistema software "Comics Store" secondo gli obiettivi espressi al punto 1.2.

2 Descrizione generale

2.1 Prospettive del prodotto

Il sistema software "Comics Store BE" è la parte centrale di un'applicazione web (Comics Store per l'appunto), rappresenta il componente che implementa la logica di business e che deve fornire un'interfaccia al componente client dell'applicazione (il Front-End). L'interfaccia menzionata sarà una API REST. Inoltre il sistema deve interagire con una base di dati che si occupa della memorizzazione e del reperimento dei dati.

2.2 Funzionalità del prodotto

Il sistema "Comics Store BE" deve:

- Fornire la ricerca di collezioni di fumetti;
- Permettere solo agli utenti registrati di:
 - aggiungere fumetti al carrello;
 - rimuovere fumetti dal carrello;
 - acquistare i fumetti presenti nel carrello;
 - creare liste desideri;
 - eliminare liste desideri;
 - aggiungere fumetti ad una lista desideri;
 - rimuovere fumetti da una lista desideri;
 - modificare alcune delle suo informazioni personali;
- Permettere solo all'amministratore di:
 - aggiungere nuove collezioni;
 - eliminare collezioni esistenti;
 - creare nuove categorie;
 - eliminare delle categorie;
 - aggiungere delle categorie ad una collezione;
 - rimuovere delle categorie da una collezione;
 - aggiungere copie di fumetti al catalogo;

- aggiungere nuovi fumetti al catalogo;
 - creare nuovi sconti;
 - aggiungere uno sconto a dei fumetti;
 - generare report delle vendite;
- Salvare tutti gli acquisti effettuati dagli utenti;

2.3 Caratteristiche utente

Il sistema software "Comics Store BE" è rivolto agli sviluppatori dell'interfaccia client che faranno uso dei servizi forniti da questo sistema.

2.4 Vincoli generali

Riguardo all'inserimento nel catalogo di fumetti e collezioni bisogna garantire che un fumetto appartenga ad *una ed una sola* collezione e che una collezione appartenga ad *almeno* una categoria.

Bisogna verificare che un utente non aggiunga più copie di un fumetto nel suo carrello di quante siano disponibili.

Prima di procedere all'acquisto dei prodotti contenuti nel carrello v'è verificato che per ognuno ci siano abbastanza copie disponibili.

3 Requisiti di interfaccia esterna

3.1 Interfaccia utente

Il sistema software "Comics Store BE" deve essere dotato di un'interfaccia REST.

3.2 Interfaccia hardware

Il sistema software "Comics Store BE" non deve interfacciarsi con nessun sistema hardware.

3.3 Interfaccia software

Il sistema software "Comics Store BE" deve interfacciarsi con una base di dati per il salvataggio ed il reperimento dei dati, è bene che si sia un livello di astrazione tra i due componenti in modo tale che si possa cambiare DBMS senza alterare il funzionamento del sistema.

È anche necessario interfacciarsi con la piattaforma di pagamento Paypal che farà da intermediario durante l'acquisto.

3.4 Interfaccia di comunicazione

Il sistema software "Comics Store BE" deve comunicare con il lato client attraverso il protocollo HTTPS.

Il sistema deve comunicare con gli utenti anche via e-mail.

4 System Features

4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1: REQ-2:

4.2 System Feature 2 (and so on)

5 Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6 Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

6.1 Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

6.2 Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

6.3 Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>