
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	5
2	Descrizione generale	6
2.1	Prospettive del prodotto	6
2.2	Funzionalità del prodotto	6
2.3	Caratteristiche utente	7
2.4	Vincoli generali	7
2.5	Assumptions and Dependencies	7
3	External Interface Requirements	8
3.1	User Interfaces	8
3.2	Hardware Interfaces	8
3.3	Software Interfaces	8
3.4	Communications Interfaces	8
4	System Features	9
4.1	System Feature 1	9
4.1.1	Description and Priority	9
4.1.2	Stimulus/Response Sequences	9
4.1.3	Functional Requirements	9
4.2	System Feature 2 (and so on)	9
5	Other Nonfunctional Requirements	10
5.1	Performance Requirements	10
5.2	Safety Requirements	10
5.3	Security Requirements	10
5.4	Software Quality Attributes	10
5.5	Business Rules	11
6	Other Requirements	12
6.1	Appendix A: Glossary	12
6.2	Appendix B: Analysis Models	12
6.3	Appendix C: To Be Determined List	12

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.

L'amministratore (il gestore della fumetteria) deve poter:

- aggiungere (e rimuovere, se necessario) nuove collezioni di fumetti specificandone il nome e l'immagine di copertina;
- creare (e rimuovere) nuove categorie di fumetti specificandone il nome ed una breve descrizione;
- associare le collezioni alle categorie, una collezione deve far parte di almeno una categoria, in generale può far parte di più categorie;
- aggiungere delle copie degli albi alle varie collezioni, si tenga presente che ogni fumetto deve far parte di una ed una sola collezione. Per aggiungere un nuovo fumetto nel catalogo l'amministratore deve specificare il numero dell'albo, il prezzo unitario, lo scrittore ed il disegnatore, il numero di pagine, il codice ISBN ed una descrizione;
- creare nuovi sconti specificando la percentuale di sconto e la data di scadenza;
- applicare uno sconto a più fumetti (non necessariamente appartenenti alla stessa collezione);
- richiedere la generazione di report indicanti, in 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.

L'utente registrato che ha effettuato l'accesso deve poter:

- aggiungere fumetti al suo carrello a patto che questi siano attualmente disponibili per l'acquisto. Se esce dall'applicazione (effettua il log out) il contenuto del carrello deve essere memorizzato;
- rimuovere fumetti dal suo carrello;
- acquistare i fumetti contenuti nel suo carrello (se disponibili nel catalogo) attraverso la piattaforma Paypal, al termine dell'operazione il carrello deve essere svuotato;
- creare (ed eliminare, se già presenti) delle liste desideri. Ad ogni lista l'utente deve poter aggiungere e rimuovere qualunque fumetto (anche fumetti che, in quel momento, non sono disponibili per l'acquisto);
- abilitare le notifiche per ogni lista desideri da lui creata: ogni cambiamento riguardante prezzo, sconto e disponibilità di ogni fumetto contenuto nella lista gli deve essere notificato via email;
- 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 deve fornire un'interfaccia al componente client dell'applicazione (il Front-End). L'interfaccia menzionata sarà una API RESTful. Inoltre deve integrarsi con un DBMS 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 sue 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

Il sistema deve rispettare i seguenti vincoli:

- Un fumetto deve far parte di una ed una sola collezione
- Una collezione deve appartenere ad almeno una categoria
- Un utente non può aggiungere al carrello più copie di un fumetto di quante ce n'è siano disponibili
- Un utente non deve concludere l'acquisto se un fumetto non è più disponibile al momento dell'acquisto

2.5 Assumptions and Dependencies

Il sistema software "Comics Store" dovrà essere utilizzato

3 External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4 System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

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.>