

BoomToys
System Desing Document
Versione 1.0



Data: 19/11/16

Partecipanti:

Nome	Matricola
D'Amato Valentina	0512103052
Russo Daniele	0512103196
Cicchelli Marco	0512103292
Sergio Massimo	0512103070

Scritto da:	D'Amato, Russo, Cicchelli, Sergio
--------------------	-----------------------------------

Revision History

Data	Versione	Descrizione	Autore
19/11/2016	0.1	Creazione documento e stesura punti: 1.1, 1.2, 1.3, 1.4, 1.5;	D'Amato V.
21/11/2016	0.3	Stesura punti: 3.1, 3.2, 3.7;	Sergio M.
22/11/2016	0.4	Stesura punti: 3.4, 3.5, 3.6, 3.7;	Daniele R.
23/11/2016	0.5	Stesura punti: 2, 4;	Cicchelli M.
25/11/2016	1.0	Revisione completa del documento ed eventuali correzioni.	D'Amato, Sergio, Russo, Cicchelli

Indice

1. Introduzione	4
1.1 Scopo del sistema	4
1.2 Obiettivi di progettazione	4
1.3 Definizioni, acronimi e abbreviazioni	4
1.4 Riferimenti	5
1.5 Overview	5
2. Architettura software attuale	5
3. Architettura software proposta	6
3.1 Panoramica	6
3.2 Decomposizione in sottosistemi	7
3.3 Mapping hardware/software	7
3.4 Gestione dei dati persistenti	8
3.5 Controllo e sicurezza degli accessi	8
3.6 Software di controllo globale	8
3.7 Boundary condition	9
4. Subsystem services	9

1 Introduzione

1.1 Scopo del Sistema

Il Progetto del sito E-commerce BoomToys ci è stato commissionato per permettere ai consumatori un acquisto facile e comodo da casa e per aumentare i profitti dell'azienda, che ci ha contattato a tale scopo. Il sito, progettato per l'azienda BoomToys, permette l'acquisto di giocattoli di varie tipologie e marche, previa registrazione e successivo login ad esso.

Il sistema andrà a velocizzare le procedure di acquisto dei vari prodotti suddivisi in gruppi per marche, oltre a permettere la consultazione dei prodotti in catalogo, corrispondenti a quelli che avrebbero potuto trovare in negozio. Questo per permettere la diffusione di un sistema di acquisto veloce e pratico accessibile davvero a tutti. Ne consegue che, il software dovrà essere rivolto a clienti di ogni età e esperienza a livello informatico; all'admin, poi, sarà consentito di aggiornare e visualizzare i prodotti all'interno del proprio magazzino e di accedere alla lista dei clienti presenti nel database che hanno effettuato la registrazione.

Secondo specifiche, il sistema dovrà permettere il salvataggio dei prodotti all'interno del carrello virtuale, il sistema provvederà ad aggiornarlo man mano che ne vengono selezionati di nuovi. Quando l'acquisto va a buon fine, deve essere possibile visualizzarne la "fattura".

1.2 Obiettivi di progettazione

Il sito BoomToys è stato sviluppato per semplificare le procedure di acquisto e consultazione dei prodotti dell'azienda. Grazie ad interfacce intuitive è possibile interagire ed amministrare il sistema senza difficoltà alcuna.

Il sistema si appoggia ad un DBMS esterno, sfruttando al massimo le potenzialità di ottimizzazione per il recupero dei dati. I prodotti della BoomToys sono acquistabili dal sito, solo dopo essersi autenticati; il database controlla l'accesso al sistema tramite l'inserimento di nome utente e password scelte dal cliente stesso, al momento della registrazione. A ogni dipendente dell'azienda che dovrà accedervi per effettuare lavori di ripopolamento saranno affidate username e password fissate dai programmatori.

Il progetto è stato realizzato nel linguaggio di programmazione Java, con paradigma di programmazione ad oggetti. La scelta di questo linguaggio di programmazione è scaturita dalla necessità di servlet che permettano la comunicazione tra client e server.

1.3 Definizioni acronimi e abbreviazioni

In questo sotto-paragrafo si specificano i termini del dominio del discorso del progetto realizzato. Essi, infatti, pur essendo di uso comune, potrebbero indurre a interpretazioni personali, quindi potenzialmente diverse da quelle sottintese in questa trattazione.

- Acquisto – inserimento nel carrello e pagamento dei/I prodotti/o.
- Prodotti – merce presente all'interno del magazzino.
- Sessione di lavoro – accesso al sistema mediante autenticazione e svolgimento delle operazioni desiderate che termina con la disconnessione dal sistema.

❖ Acronimi e abbreviazioni

- DB – DataBase
- DBMS – DataBase Management System
- GUI – Graphical User Interface
- JDK – Java Development Kit
- UML – Unified Modeling Notation
- JVM – Java Virtual Machine
- JDBC – Java DataBase Connectivity
- XML – eXtensible Markup Language

1.4 Riferimenti

Per la realizzazione del sistema sono stati utilizzati i seguenti materiali di riferimento:

- EclipseJavaEE
- Apache TomCat 8.0

1.5 Panoramica

1 Introduzione

- 1.1 Scopo del Sistema
- 1.2 Obiettivi di progettazione
- 1.3 Definizioni, acronimi e abbreviazioni
- 1.4 Riferimenti
- 1.5 Panoramica

2 Architettura software attuale

3 Architettura software proposta

- 3.1 Panoramica
- 3.2 Decomposizione in sottosistemi
- 3.3 Mapping hardware/software
- 3.4 Gestione dei dati persistenti
- 3.5 Controllo e sicurezza degli accessi
- 3.6 Software di controllo globale
- 3.7 Boundary Conditions

4 Subsystem services

2 Architettura software attuale

Il sistema che intendiamo realizzare verrà ideato da principio, quindi non sarà un'opera di reingegnerizzazione. Il sistema dovrà automatizzare tutte le operazioni inerenti agli acquisti di giocattoli, che al momento l'azienda svolge nel modo tradizionale. Allo stato attuale, ad esempio, gli utenti provvedono ad acquistare i prodotti andando fisicamente in negozio prendendo la merce desiderata e recandosi in cassa per pagare. La procedura attuale risulta poco efficiente per l'azienda, poiché un utente che non riesce a raggiungere fisicamente il negozio per acquistare quel determinato prodotto, cercherà altrove.

3 Architettura software proposta

Il sistema proposto ha l'obiettivo di migliorare la situazione attuale della gestione degli acquisti da più punti di vista (utente e amministratore). In particolar modo con l'impiego di questo sistema, gli utenti effettueranno gli acquisti in modo più semplice e sicuro, grazie ad un accesso al sistema tramite username e password, e veloce, visto che gli utenti avranno immediatamente un catalogo completo dei prodotti per marche tra cui scegliere. Inoltre l'utente potrà effettuare altre operazioni quali cancellazione inserimento prodotto/i nel carrello o visualizzare i propri acquisti. Il sistema supporterà anche la figura dell'amministratore nell'aggiornamento catalogo merce (inserimento prodotti) e nella visione degli utenti che si sono iscritti al sito.

3.1 Panoramica

Il sistema che si intende realizzare è un sistema distribuito in cui ogni utente, in seguito a registrazione di un account, potrà autenticarsi ed usufruire delle sue funzionalità.

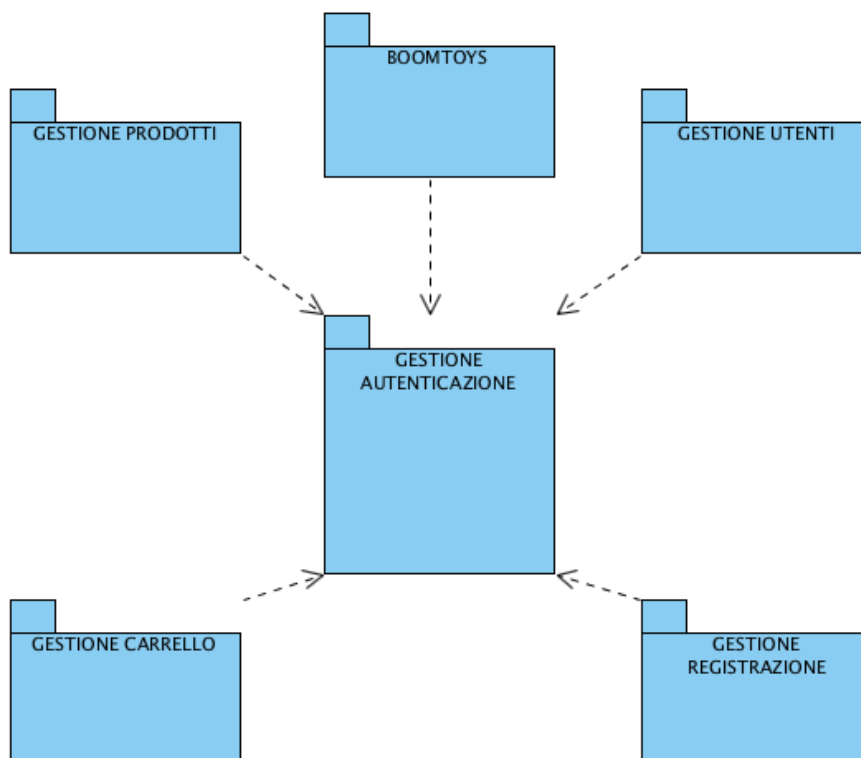
il sistema sarà fruibile da qualsiasi terminale collegato a rete web.

L'architettura di sistema proposta è basata sul pattern architetturale MVC (Model, View, Controller).

Si identificano tre livelli logici:

- Le view: includono gli oggetti boundary, le form e tutte le interfacce grafiche che interagiranno in modo diretto con l'utente;
- I controller: ricevono la richiesta dall'utente (inviata tramite le view), effettua dei controlli sugli input, racchiude la logica applicativa del sito, specifica il flusso dell'applicazione e controlla l'interazione tra gli altri livelli;
- Model: rappresenta lo stato dell'applicazione, specifica le procedure di estrazione da DB ed elaborazione dei dati;
- Lo storage del sistema sarà gestito da un DBMS che si occuperà di recuperare, memorizzare ed interrogare i dati presenti nel database, elaborando le richieste degli utenti.

3.2 Decomposizione in sottosistemi



3.3 Mapping hardware/software

Il sistema che sarà realizzato si basa su un'architettura Web-based.

Protocollo Richiesto: HTTP

Memorizzazione Dati: DMBS MySQL

WebServer: ApacheTomcat 8.0

Linguaggi di programmazione utilizzati: Html, CSS, Javascript, Servlet.

Web Server

Il web server utilizzato è ApacheTomcat, compatibile con i più diffusi sistemi operativi (Windows, Linux e Mac).

View

Dipende direttamente dal web server, per cui viene usato HTML insieme a codice javascript lato client.

Controller

Tutte le funzionalità sono implementate in linguaggio Java. Le Jsp interrogano le servlet che inviano la risposta in HTML al web server e quest'ultimo invia al browser client il codice HTML.

Model

Si interfaccia con il database server, adempiendo alle richieste del controller che necessitano il trattamento di informazioni memorizzate nel database. Utilizza le funzioni di interrogazione che permettono l'interfacciamento con MySQL.

Database Server

Il DBMS usato è MySQL, che si interfaccia benissimo con il web server Apache.

3.4 Gestione dei dati persistenti

Il sito realizzato è stato progettato e si avvale dell'uso di un database di tipo relazionale, al quale si interfaccia tramite un DBMS. Questa scelta è stata ponderata dal bisogno di avere un database che garantisca una sicurezza dei dati sensibili, assieme ad una maggiore affidabilità e garanzia di coerenza e facilità di gestione, nonché dalla velocità di accesso e trasmissione dei dati.

3.5 Controllo e sicurezza degli accessi

Il nostro sito è basato su un sistema di tipo multi-utente. Qui troviamo diversi attori che hanno accesso alle diverse funzionalità e informazioni del sistema. Quindi è stata definita una politica di controllo degli accessi e di sicurezza, che identifichi le operazioni e i privilegi di ogni attore all'interno del sistema. Questo deve essere fatto affinché si eviti il malfunzionamento e l'utilizzo di sessioni di sistema riservati ad altri attori del sistema.

Il sistema tramite la combinazione di username e password fornita ad ogni utente, regola gli accessi, il sistema prenderà questi due valori e li confronterà con quelli all'interno del database, se entrambi risultano corretti, non ci saranno problemi d'accesso altrimenti verrà negato l'accesso e verrà presentato all'utente un messaggio di errore e riproposto l'inserimento dei dati necessari per l'accesso. Discorso analogo per quanto riguarda l'amministratore a cui verranno forniti username e password standard affinché possa accedere al portale per eventuali modifiche.

Il sistema prevede due tipologie di utenti: Amministratore e Utente Registrato.

- Amministratore:** entra nel sistema tramite l'uso di username e password (affidategli dai programmatori), gestisce il catalogo generale dei prodotti, potendoli inserire e cancellare a sua discrezione, visualizza gli utenti registrati.

- Utente Registrato:** Accede al sistema tramite l'uso di un username e password scelte da lui, effettua l'acquisto dei prodotti, può controllare gli acquisti pregressi.

3.6 Software di controllo globale

Per quanto riguarda il flusso di controllo esterno (fra sottosistemi), il server sarà in funzione in locale solo quando quest'ultimo viene avviato per ovviare alle richieste di servizi da parte dell'utenza ed eventuali modifiche da parte dell'amministratore; nel caso di più richieste contemporanee, il server utilizzerà la tecnica FIFO (First In First Out).

3.7 Boundary condition

1. Il server deve essere sempre in funzione altrimenti i clienti o l'amministratore non possono richiedere funzionalità quali l'autenticazione e di conseguenza gli eventuali acquisti o modifiche da parte di quest'ultimi, l'inserimento o la cancellazione di un prodotto all'interno del catalogo, la piena visione degli acquisti da parte dell'utenza.
2. In presenza di una chiusura o malfunzionamento del sito, il sistema funziona correttamente ma non si potrà usufruire delle funzioni presenti ma solo tramite il database che gestisce quest'ultimo.
3. In caso di errata digitazione di username e/o password da parte di un utente o un amministratore durante l'accesso, il sistema non permette l'accesso al portale mostrando un messaggio d'errore. Di conseguenza l'utenza potrà riprovare digitando i dati nuovamente nella schermata di login.

4. Subsystem services

1. Il sottosistema "Gestione Autenticazione" si occupa di effettuare il login e il logout per gli utenti, con i relativi controlli di accesso;

Le funzioni principali del model sono:

- Login: permette agli utenti di loggarsi al sistema;
- Logout: permette all'utente di terminare la sessione e di disconnettersi dal sistema;

2. Il sottosistema "Gestione Registrazione" invece si occupa della registrazione degli utenti; La principale funzione è:

- Registrazione: permette agli utenti di registrarsi al Sistema;

3. Il sottosistema "Gestione Prodotti" riguarda tutte le operazioni che è possibile effettuare sui prodotti e sul catalogo dei prodotti stessi;

Le funzioni principali del controller sono:

- Inserire prodotto: permette all'amministratore di inserire un prodotto nel catalogo;
- Cancella prodotto: permette all'amministratore di cancellare un prodotto dal catalogo;
- VisualizzaCatalogo: permette di visualizzare la lista completa dei prodotti disponibili nel catalogo;

4. Il sottosistema "Gestione Carrello" riguarda tutte le operazioni che è possibile effettuare sul carrello per il successivo acquisto dei prodotti.

Le funzioni principali sono divise in tre categorie:

➤ Servizi acquisti:

- Acquista prodotto: permette ad un utente registrato di acquistare un prodotto

inserito nel carrello.

- Storico acquisto:
 - Visualizza “I miei acquisti”: permette ad un utente di visualizzare tutti i prodotti acquistati in base alla data, quantità, prezzo e IBAN;
 - Contenuto carrello:
 - Inserisci: è una funzione che viene eseguita ogni volta che un utente vuole inserire un nuovo prodotto all'interno del carrello;
 - Rimuovi: è una funzione che viene eseguita ogni volta che un utente decide di eliminare un prodotto all'interno carrello.
5. Il sottosistema “Gestione Utenti” riguarda la gestione amministrativa dei vari utenti;
Le funzioni principali sono:
- Visualizza “Info utenti”: E’ una funzione che permette di visualizzare tutti le informazioni riguardanti gli utenti.
 - Cancella utente: Permette di eliminare un utente dall’elenco.