



Università degli Studi di Ferrara
Dipartimento di Ingegneria

Corso di Laurea Triennale in Ingegneria Elettronica
e Informatica

Documentazione Ingegneria del Software

Progetto E-Commerce: BINS

Studente:
Michele Vaccari
Matricola 121955

Docente:
Prof. Fabrizio Luglio

Anno Accademico 2016–2017

INDICE

1	SPECIFICHE PROGETTO	1
1.1	Caratteristiche del sistema	1
1.2	Utenti del sistema	2
2	MODELLO DEI CASI D'USO	3
2.1	Diagramma dei casi d'uso	3
2.1.1	Caso d'uso: Visualizza pagine statiche	3
2.1.2	Caso d'uso: Effettua registrazione	4
2.1.3	Caso d'uso: Effettua login	5
2.2	Diagramma delle attività	6

ELENCO DELLE FIGURE

Figura 1	Attore utente pubblico	3
----------	------------------------	---

INTRODUZIONE

BINS (acronimo ricorsivo di "BINS Is Not Shopping") è un applicazione web liberamente disponibile, indicata soprattutto per commercializzare prodotti alimentari. Lo scopo di questo lavoro è di documentare l'applicazione usando l'UML (Unified Model Language).

La documentazione è articolata come segue.

IL PRIMO CAPITOLO contiene le specifiche del progetto descrivendo le caratteristiche e le categorie di utenti dell'applicazione.

IL SECONDO CAPITOLO descrive che cosa deve fare il sistema e quali sono gli attori per poterlo far funzionare.

Per il *Presentation Level* si è usato il framework *Bootstrap* sfruttando al meglio il *responsive design* che viene messo a disposizione dal framework. Per quanto riguarda l'usabilità e il design si è scelto l'approccio *mobile first* ma si è anche prestato attenzione all'uso dell'applicazione in modalità desktop.

Per l'application server si è usato *Tomcat* e si è utilizzato il linguaggio di programmazione Java. Per interfacciare l'application server con il database si è utilizzato JDBC.

Il DBMS utilizzato è *MySQL*.

1.1 CARATTERISTICHE DEL SISTEMA

Si vuole progettare un'applicazione web per la vendita on line dei prodotti di un supermercato. L'applicazione deve avere un frontend multilingue (italiano e inglese).

I prodotti del supermercato sono suddivisi in reparti (ortofrutta, macelleria, accessori per la casa, cura della persona, ecc.) e ogni prodotto può avere una data di scadenza.

L'applicazione web deve avere le seguenti caratteristiche:

1. Possibilità di visualizzare il catalogo dei prodotti, navigabile per reparto, caratteristiche, marchio, data di scadenza, ricerca libera, ecc. Possibilità di vedere il singolo prodotto con tutti i dettagli;
2. Possibilità di inserire prodotti nel carrello e di effettuare l'acquisto di più prodotti in diverse quantità. Predisporre la gestione dei prezzi, del totale carrello, la gestione della disponibilità di magazzino, impedendo di poter acquistare quantità non disponibili e prodotti scaduti;
3. Gestione di una o più shopping list (lista della spesa);
4. Possibilità di inserire o selezionare un indirizzo di consegna da una rubrica personale, di simulare il pagamento e di confermare l'ordine;
5. Possibilità di visualizzare lo stato dell'ordine e lo storico degli ordini effettuati;
6. Gestione di coupon o buoni sconto in fase di acquisto, possibilità di visualizzare il tracking dell'ordine in consegna;
7. Possibilità di gestire il catalogo (inserimento, modifica, blocco prodotti), i reparti, i brand, ecc. Possibilità di gestire la disponibilità di magazzino;
8. Gestione di prodotti in push (per cui spingere la vendita) con inserimento in una "vetrina in home page" o in un'area promo;
9. Possibilità di visualizzare gli utenti, verificare il numero degli ordini per utente, bloccare eventuali utenti, gestire gli altri utenti amministratori;
10. Gestione dei coupon o buoni sconto;

11. Gestione del tracking dell'ordine, con simulazione di tutti i cambi di stato (consegnato al corriere, in viaggio, consegnato al destinatario, ecc.);

1.2 UTENTI DEL SISTEMA

Il sistema prevede che le categorie di utenti sia così rappresentata:

UTENTI PUBBLICI Possono effettuare solo il punto [1](#) ed eventualmente registrarsi.

UTENTI REGISTRATI Possono effettuare i punti dal [1](#) al [6](#) compresi.

AMMINISTRATORI Possono effettuare i punti dal [7](#) al [11](#) compresi.

2 | MODELLO DEI CASI D'USO

2.1 DIAGRAMMA DEI CASI D'USO

Si esegue l'analisi dei casi d'uso nel dominio degli attori che interessano i casi d'uso medesimi suddividendo lo schema completo in sottoschemi, uno per ogni attore, al fine di mantenere una maggior leggibilità del progetto. Si documenta ogni caso d'uso al fine di descrivere nel dettaglio il comportamento del sistema senza riferirsi ad una particolare implementazione.

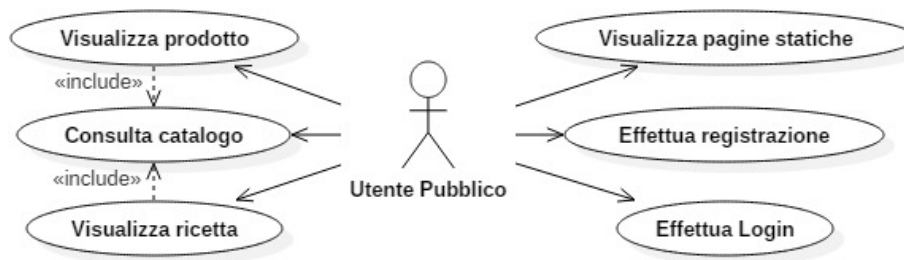


Figura 1: Attore utente pubblico

2.1.1 Caso d'uso: Visualizza pagine statiche

Descrizione

Questa funzionalità permette all'utente pubblico di visualizzare tutte le pagine richieste con contenuto non dinamico.

Attori coinvolti

Utente pubblico, partecipazione attiva, richiedendo la visualizzazione delle pagine.

Pre-condizioni

Nessuna preconditione, in quanto non serve uno stato particolare per visualizzare una pagina statica.

Post-condizioni

Nessuna postcondizione, in quanto la visualizzazione di una pagina statica non altera lo stato dell'applicazione.

Flusso principale

1. Richiesta di qualsiasi pagina che abbia contenuto statico da parte dell'utente pubblico
2. Risposta del server con la pagina statica richiesta da parte dell'utente pubblico;

Flussi alternativi

Non presenti.

2.1.2 Caso d'uso: Effettua registrazione**Descrizione**

Questa funzionalità permette di registrare l'utente all'interno del database, consentendogli tutte le operazioni dell'utente registrato.

Attori coinvolti

Utente pubblico, partecipazione attiva dell'attore verso il caso d'uso medesimo.

Pre-condizioni

Nessuna preconditione

Post-condizioni

Viene aggiornato lo stato sul database con l'inserimento di un nuovo utente.

Flusso principale

1. L'utente visualizza il form da compilare per effettuare la registrazione nel sistema;
2. L'utente compila tutti i campi visualizzati nel modulo con controllo diretto se i dati inseriti sono sintatticamente corretti oppure no;
3. L'utente capisce che il dato inserito è sbagliato se il campo dove ha inserito l'informazione diventa di colore rosso, altrimenti se il colore è inalterato vuol dire che dal punto di vista sintattico le informazioni sono corrette;
4. Finché il modulo non è stato inviato al server l'utente può modificare i dati inseriti negli appositi campi di compilazione;
5. Quando l'utente richiede l'invio dei dati al server vengono controllati nuovamente tutti i campi del modulo, inoltre è richiesta l'accettazione delle condizioni di utilizzo;

6. Una volta che i dati sono stati inviati al server vengono elaborati e memorizzati sul database modificando quindi lo stato del database stesso;
7. Viene notificato all'utente l'avvenuto inserimento del proprio profilo nel sistema.

Flussi alternativi

Nel caso di inserimento di un indirizzo email già esistente nel database, dopo il punto 5 viene visualizzato nuovamente il form precompilato con i dati inseriti nell'ultima compilazione del form, notificando l'utente della presenza nel database di un profilo con lo stesso indirizzo email.

2.1.3 Caso d'uso: Effettua login

Descrizione

Questa funzionalità permette all'utente pubblico di farsi riconoscere dal sistema, accedendo quindi a tutte le sue funzionalità.

Attori coinvolti

Utente pubblico, partecipazione attiva dell'attore che passa da *Utente Pubblico* a *Utente registrato* o *Utente amministratore* o *Utente super amministratore*.

Pre-condizioni

Compilazione del modulo di login.

Post-condizioni

L'utente passa da attore *Utente Pubblico* ad attore *Utente registrato* o ad attore *Utente amministratore* o ad attore *Utente super amministratore*.

Flusso principale

1. Il form per l'autenticazione è visibile in tutte le pagine del sito;
2. L'utente pubblico compila il modulo inserendo la propria email e la propria password;
3. Il sistema effettua verifica se le informazioni inserite nel campo email sono corrette dal punto di vista sintattico, colorando il campo di colore rosso se l'indirizzo inserito è sbagliato dal punto di vista sintattico;
4. L'utente clicca sul pulsante per l'autenticazione e vengono inviati i dati al server;
5. Dopo aver ricevuto i dati il server controlla se nella base dati è presente un utente con indirizzo email e password ricevuti;

6. Se le informazioni esistono sulla base dati l'utente pubblico viene riconosciuto dal sistema e al posto del form per l'autenticazione viene visualizzato il nome dell'utente e un menù con tutte le operazioni che l'utente può eseguire sul sistema.

Flussi alternativi

Nel caso di inserimento di email e password non presenti nel database vengono notificate all'utente pubblico le possibili cause di errore.

2.2 DIAGRAMMA DELLE ATTIVITÀ

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.