

Documento di analisi

Vista statica dell'interfaccia

VisualizzaProdotto

Utente
matteogambella@hotmail.com

Sito Abbigliamento Sportivo

Ordini

Carrello Spesa

Menù di ricerca del prodotto

Sesso

Uomo


Categoria

T-Shirt

Marca

Gymshark

CERCA



Altezza modello :
1.85m

Taglia modello:
L

GUIDA ALLE TAGLIE

Taglia	Petto(Cm)	Vita(Cm)	Fianchi(cm)
S	100	79	90
M	105	84	95
L	110	89	100
XL	115	94	105
XXL	120	99	110

Divide T-shirt

Nome del prodotto

Gymshark

Marca

25.00 euro

Prezzo

Voto medio degli utenti: 3.0

COLORI

rosso

verde

blu

SELEZIONA TAGLIA

S

M

L

XL

XXL

QUANTITÀ

1

AGGIUNGI AL CARRELLO

Caso d'uso

- 1.L'utente riempie i campi testo preposti alla ricerca del prodotto(sesso,categoria e marca) e poi schiaccia il pulsante cerca.
- 2.Il Sistema visualizza una tabella con tutte le immagini dei prodotti che soddisfano la ricerca e un grafico a torta che mostra gli N prodotti più acquistati in percentuale (sempre che soddisfano i parametri già citati).
- 3.L'utente schiaccia l'immagine del prodotto.

4. Il sistema visualizza le informazioni che riguardano il prodotto e i controlli per l'acquisto del prodotto.

5. L'utente seleziona il colore, la taglia e la quantità del prodotto e preme aggiungi carrello.

6. Il sistema visualizza un messaggio che notifica che il prodotto è stato aggiunto o meno al carrello e aggiorna il carrello spesa in caso tutto sia andato a buon fine.

File di configurazione XML

All'avvio il Sistema legge dal file di configurazione i seguenti dati:

- username dell'utente che sta usando l'applicazione;
- numero di prodotti con più acquisti percentuali da mostrare nel grafico a torta
- numero di giorni da considerare nella valutazione degli acquisti
- la capienza massima del carrello
- ip e porta del server di log
- ip e porta del server DBMS

Cache locale degli input

Alla chiusura il sistema salva gli articoli presenti nel carrello e i valori dei controlli (campi testo, radiocomandi...) della vista in corso (seleziona prodotto, storico degli ordini, carrello spesa...). All'apertura del sistema se presenti su file binario vengono ripristinati gli articoli del carrello spesa e i valori dei controlli salvati alla precedente chiusura. Si riparte dall'ultima vista visualizzata.

Archivio

Il sistema archivia per ogni ordine:

- l'username del cliente
- il codice dell'ordine
- la data di invio dell'ordine
- il nome della spedizione
- l'indirizzo di spedizione
- il cap
- la città di spedizione

Il sistema archivia per ogni articolo dell'ordine:

- il codice dell'ordine

- il nome del prodotto
- il colore selezionato
- la taglia selezionata
- il voto(inizialmente null)

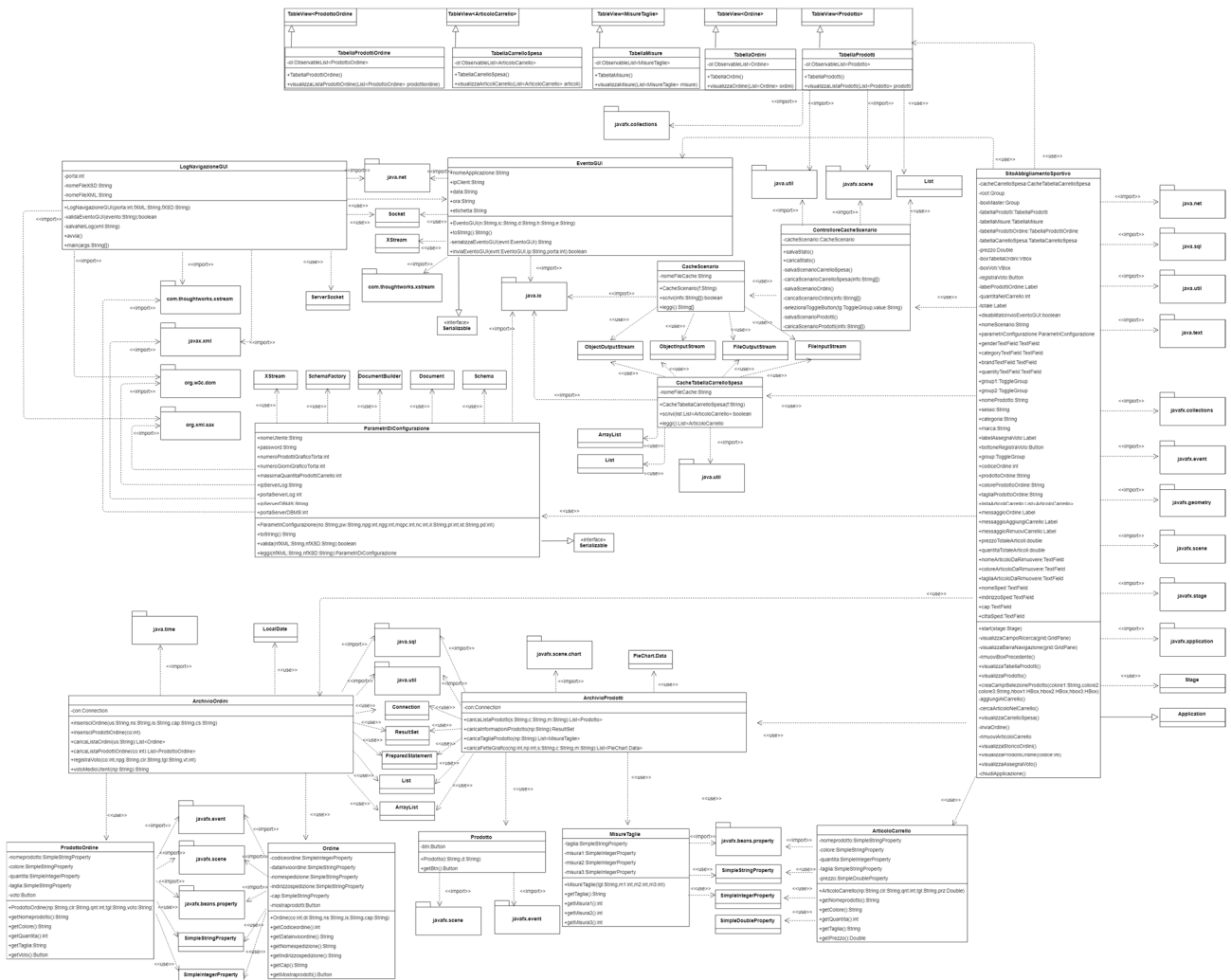
File di log remoto XML

Il sistema invia una riga di log ad ogni evento di seguito:

- Avvio dell'applicazione ("AVVIO")
- Ricerca di un determinato tipo di prodotto("CERCA")
- Pressione pulsante di selezione del prodotto("SELEZIONE PRODOTTO")
- Pressione pulsante di aggiungi al carrello ("AGGIUNGI AL CARRELLO")
- Pressione pulsante di visualizzazione del carrello ("CARRELLO SPESA")
- Pressione pulsante di rimozione di un articolo dal carrello ("RIMUOVI ARTICOLO")
- Pressione pulsante di invio dell'ordine ("INVIA ORDINE")
- Pressione pulsante di visualizzazione dello storico ordini ("ORDINI")
- Pressione pulsante di visualizzazione dei prodotti relativi all'ordine ("MOSTRA PRODOTTI ORDINE")
- Pressione pulsante per mostrare il controllo per l'assegnazione del voto al prodotto ("ASSEGNA VOTO PRODOTTO")
- Pressione pulsante di registrazione del voto espresso ("REGISTRA VOTO")
- Termine dell'applicazione ("TERMINE")

La riga di log contiene: nome dell'applicazione, indirizzo IP del client, data e ora corrente, l'etichetta associata all'evento.

Diagramma delle classi



Responsabilità delle classi:

- **SitoAbbigliamentoSportivo** : classe controller che estende `javafx.application.Application`. Proprio perché ha il metodo `start` ha il compito di inizializzare l'interfaccia. Inoltre contiene metodi per la gestione del carrello spesa, il metodo per l'inoltro dell'ordine e permette la navigazione tra le diverse viste del progetto (prodotti, ordini e carrello spesa).
- **ProdottoOrdine**: classe bean utilizzata per inserire o leggere i prodotti relativi a un ordine dal database.

- Ordine: classe bean utilizzata per inserire o leggere gli ordini relativi a un dato utente da database.
- Prodotto: classe utilizzata per realizzare i bottoni della tabella dei prodotti, ovvero le immagini cliccabili.
- MisureTaglie: classe bean utilizzata per leggere le misure delle taglie relative a un dato prodotto da database.
- ArticoloCarrello: classe bean utilizzata per inserire o rimuovere articoli dal carrello spesa.
- TabellaProdottiOrdine: classe che estende TableView<ProdottoOrdine>, si occupa di inizializzare la tabella che contiene i prodotti di un determinato ordine e contiene il metodo che permette di aggiornarla.
- TabellaOrdini: classe che estende TableView<Ordine>, si occupa di inizializzare la tabella che contiene gli ordini di un determinato utente e contiene il metodo che permette di aggiornarla.
- TabellaProdotti: classe che estende TableView<Prodotto>, si occupa di inizializzare la tabella che contiene i prodotti che soddisfano una determinata ricerca e contiene il metodo che permette di aggiornarla.
- TabellaMisure: classe che estende TableView<MisureTaglie>, si occupa di inizializzare la tabella che contiene le misure delle taglie relative a un determinato prodotto e contiene il metodo che permette di aggiornarla.
- TabellaCarrelloSpesa: classe che estende TableView<ArticoloCarrello>, si occupa di inizializzare la tabella che contiene gli articoli del carrello di un determinato utente e contiene il metodo che permette di aggiornarla.
- ArchivioOrdini permette di effettuare operazioni sul database di inserimento/lettura degli ordini, inserimento/lettura di prodotti di un ordine, registrazione del voto degli utenti riguardo un determinato prodotto e la lettura della media dei voti degli utenti riguardo un prodotto.
- ArchivioProdotti permette di effettuare operazioni sul database di lettura di prodotti che soddisfano dei parametri, lettura di informazioni sui prodotti (come ad es. la possibile gamma di colori tra cui scegliere), lettura dei valori delle taglie relative a un prodotto, calcolo dei prodotti con più acquisti percentuali che soddisfano certe condizioni.
- ParametriDiConfigurazione: classe serializzabile che conterrà le configurazioni dell'applicazione lette da un file xml; inoltre ha il compito di caricare una configurazione XML leggendola da un file, dopo averlo validato rispetto ad una grammatica XSD.
- CacheTabellaCarrelloSpesa: permette di leggere e scrivere da un file binario che fungerà da cache per la tabella del carrello spesa.

- CacheScenario: permette di leggere e scrivere da un file binario che fungerà da cache per i vari controlli del progetto.
- ControlloreCacheScenario: nasce per la vastità di controlli del progetto e per avere la possibilità di poter salvare o leggere da file binario solo i valori dei controlli attivi all'ultima chiusura dell'applicazione. Per questo è un intermediario tra la cache vera e propria e la classe principale.
- EventoGUI: classe serializzabile che contiene le informazioni di un evento GUI: nome dell'applicazione, ip del client, data e ora; inoltre permette di inviare un evento gui ad un server di log, serializzandolo in XML.
- LogNavigazioneGUI: server di log per gli eventi gui. Permette di ricevere gli eventi serializzati in xml, validarli con una grammatica XSD e salvarli in append ad un file di log.

Manuale utente

Per poter eseguire l'applicazione SitoAbbigliamentoSportivo seguire i seguenti step:

1. Aprire il file parametridiconfigurazione.xml e configurare i parametri

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ParametriDiConfigurazione>
3      <nomeUtente>mattgamb@hotmail.com</nomeUtente>
4      <numeroProdottiGraficoTorta>2</numeroProdottiGraficoTorta>
5      <numeroGiorniGraficoTorta>30</numeroGiorniGraficoTorta>
6      <massimaQuantitaProdottiCarrello>60</massimaQuantitaProdottiCarrello>
7      <ipServerLog>127.0.0.1</ipServerLog>
8      <portaServerLog>4242</portaServerLog>
9      <ipServerDBMS>127.0.0.1</ipServerDBMS>
10     <portaServerDBMS>3306</portaServerDBMS>
11 </ParametriDiConfigurazione>
```

2. Lanciare l'applicazione

3. Inizialmente la schermata appare vuota, inserire nei campi di ricerca dei parametri di ricerca come quelli nell'esempio e premere "Cerca"

The screenshot shows a web browser window titled "Sito abbigliamento sportivo". The interface has a dark orange background. At the top, there is a header bar with a yellow gradient. On the left of the header, it says "Utente: mattgamb@hotmail.com". In the center, it says "Sito Abbigliamento Sportivo". On the right, there are two red buttons: "Ordini" and "Carrello Spesa". Below the header, on the left side, there is a yellow box titled "Menù di ricerca del prodotto". Inside this box, there are three input fields: "Sesso:" with the value "Uomo", "Categoria:" with the value "T-Shirt", and "Marca:" with the value "Gymshark". Below these fields is a red button labeled "Cerca". The rest of the page is currently empty.

4. Il sistema visualizza la lista di prodotti e il grafico e svuota i campi di inserimento. Se si desidera visualizzare un prodotto cliccare sull'immagine.



5. Se si vuole aggiungere il prodotto al carrello spesa basta premere su Aggiungi al carrello ovviamente dopo aver selezionato colore, taglia e quantità desiderata. Il sistema darà un messaggio di errore se il valore non è ammissibile (≤ 0), se non è

stato espresso alcun valore o se infine la disponibilità del carrello non è sufficiente a contenere la quantità desiderata.

6. Il carrello spesa verrà aggiornato come sotto. Verrà inoltre incrementata la variabile che tiene conto della capienza attuale del carrello.

Sito abbigliamento sportivo

Utente:
mattgamb@hotmail.com

Sito Abbigliamento Sportivo

Ordini Carrello Spesa

Menù di ricerca del prodotto

Sesso:

Categoria:

Marca:

Cerca

Nome spedizione:

Indirizzo spedizione:

Cap:

Città:

IN VIA ORDINE

CARRELLO SPESA

NOME PRODOTTO	COLORE	QUANTITÀ	TAGLIA	PREZZO
Divide T-shirt	rosso	1	5	25.0

Totale: 25.0

ARTICOLO DA RIMUOVERE:

NOME ARTICOLO

COLORE ARTICOLO

TAGLIA ARTICOLO

RIMUOVI ARTICOLO

7. Se si termina, poi al successivo avvio dell'applicazione il carrello spesa risulterà come prima e inoltre l'applicazione ripartirà anche con gli stessi valori inseriti nei campi di inserimento prima della chiusura.