

Contenuti

1	Manuale d'uso	2
1.1	Registrazione e Accesso	2
1.2	Lato Utente	3
1.2.1	Aggiungere e Prelevare fondi	3
1.2.2	Acquistare un prodotto	3
1.2.3	Visualizzare il Carrello e completare l'ordine	4
1.2.4	Visualizzare Notifiche delle liste di attesa	4
1.3	Lato Amministrazione	4
1.3.1	Modificare Scorte di un prodotto	5
1.3.2	Aggiungere un nuovo prodotto al catalogo	5
2	Scelte Implementative	6
2.1	Strutture Dati Utilizzate	6
2.2	Funzionalità Principali	6
2.2.1	Funzione latoUtente()	7
2.2.2	Funzioni gestisciAcquisto()	7
2.2.3	Funzione gestisciListaDiAttesa()	7
2.3	Conclusioni	9

Report Primo Progetto LASD

Davide Soldatini N86002862

Aprile 2022

1 Manuale d'uso

L'applicazione si presenta come un'interfaccia da terminale in cui, nella maggior parte dei casi, bisognerà inserire un numero oppure un carattere (s/n) per effettuare la scelta desiderata.

Per rendere l'interfaccia più pulita e meno confusionaria ho utilizzato i colori nelle printf, questi vengono riconosciuti dal terminale interno in VSCode, dal terminale Linux e dalla PowerShell di Windows ma mi hanno dato problemi quando si avvia l'applicazione direttamente dal cmd di Windows.

In ogni caso se dovessero esserci problemi ho incluso nello zip anche una **versione identica del progetto ma senza colori**.

1.1 Registrazione e Accesso

Ovviamente prima di poter utilizzare l'applicazione come utente bisogna registrarsi, occorrerà quindi selezionare "Non hai un account? Registrati" ed inserire i dati, dopodiché sarà possibile accedere.

```
-----
BENVENUTO NELLO SHOPPING ONLINE
1) Accedi come utente
2) Accedi come amministratore
3) Non hai un account? Registrati
4) Esci
3
-----
REGISTRAZIONE UTENTE
Inserisci nome utente: Utente
Inserisci password: mypass
Conferma password:
```

Per il lato amministrazione invece ho immaginato che una persona non possa registrarsi autonomamente, dato che poi potrebbe modificare liberamente la disponibilità dei prodotti. Quindi in un contesto reale ci sarà qualcuno che andrà a modificare manualmente il contenuto del file amministratori.txt. Comunque per testare il lato amministrazione potete utilizzare le credenziali: Nome amministratore: **admin**, Password: **pass**.

1.2 Lato Utente

Nella schermata principale del lato utente verrà mostrato il vostro saldo ed un menù con tutte le possibili operazioni da effettuare.

```
LATO UTENTE
Benvenuto/a, daddi
Saldo disponibile: 52.26 euro

1) Aggiungi fondi
2) Preleva fondi
3) Acquista un prodotto
4) Visualizza Carrello
5) Visualizza Notifiche liste di attesa
6) Logout
```

1.2.1 Aggiungere e Prelevare fondi

Selezionando 1) o 2) è possibile aggiungere/prelevare fondi dal proprio conto. Vi chiederà semplicemente di aggiungere un importo in euro e se tutto andrà a buon fine effettuerà l'operazione.

```
Inserire importo da aggiungere in euro (es. 49.99): 10.99
Deposito effettuato con successo!
```

```
Inserire importo da prelevare in euro (es. 49.99): 100
Il saldo non e' sufficiente per effettuare il prelievo
```

1.2.2 Acquistare un prodotto

Selezionando 3) apparirà un elenco di tutti i prodotti disponibili con il rispettivo prezzo, dopo aver fatto la vostra scelta vi verranno mostrate tutte informazioni sul prodotto desiderato e potrete acquistarne una taglia.

```
PRODOTTI DISPONIBILI
1) Maglietta Blu 10.50 euro
2) Jeans Corto 22.00 euro
3) Giubbotto Nero 45.99 euro
4) Pantalone Rosa 20.49 euro
5) Cappello Nero 10.99 euro
6) Costume Mimetico 14.99 euro
7) Camicia Bianca 37.99 euro

Selezionare il prodotto per visualizzare i dettagli: 1

INFORMAZIONI PRODOTTO
Nome: Maglietta Blu
Prezzo: 10.50 euro
Taglie S disponibili: 5
Taglie M disponibili: 7
Taglie L disponibili: 20

1)Acquista taglia S
2)Acquista taglia M
3)Acquista taglia L
4)Annulla Acquisto
1
Acquisto effettuato con successo!
```

Quando si prova ad acquistare un prodotto ma non si hanno abbastanza soldi sul conto, ovviamente viene richiesta una ricarica e il prodotto viene spostato nel **carrello**.

Quando invece si prova ad acquistare un prodotto di cui la taglia non è disponibile si viene aggiunti ad una **lista di attesa**.

1.2.3 Visualizzare il Carrello e completare l'ordine

Selezionando 4) si accede ad una semplice schermata in cui viene visualizzato il contenuto del carrello ed il prezzo totale dei prodotti.

Scegliendo "Procedi all'ordine" verrà acquistato tutto il contenuto del carrello.

```
CONTENUTO CARRELLO
- Maglietta Blu Taglia S 10.50 euro
- Giubbotto Nero Taglia M 45.99 euro
- Camicia Bianca Taglia L 37.99 euro

Prezzo totale: 94.48 euro
1) Procedi all'ordine
2) Indietro
```

1.2.4 Visualizzare Notifiche delle liste di attesa

In questa schermata verranno visualizzati tutti i prodotti per i quali l'utente si trova in una lista di attesa.

Quando uno di questi prodotti torna disponibile apparirà una notifica ma **soltamente se l'utente ha priorità massima** nella lista di attesa (si veda implementazione per i dettagli).

```
NOTIFICHE LISTE DI ATTESA
Prodotti per i quali sei in una lista di attesa:
- Pantalone Rosa Taglia S
- Maglietta Blu Taglia S
```

L'utente non ha priorità su nessun prodotto

```
NOTIFICHE LISTE DI ATTESA
Prodotti per i quali sei in una lista di attesa:
- Pantalone Rosa Taglia S
- Maglietta Blu Taglia S

Il prodotto Maglietta Blu Taglia S e' tornato disponibile!
Vuoi aggiungerlo al carrello?(s/n)
```

L'utente ha priorità su Maglietta Blu taglia
S

1.3 Lato Amministrazione

Un amministratore può incrementare il numero di taglie disponibili per un prodotto già presente oppure aggiungere un prodotto completamente nuovo.

```
LATO AMMINISTRAZIONE
Benvenuto/a, admin

1) Modifica scorte dei prodotti gia' presenti
2) Aggiungi un nuovo prodotto al catalogo
3) Logout
```

1.3.1 Modificare Scorte di un prodotto

Semplicemente, come quando si va a fare un acquisto, viene mostrata una lista di tutti i prodotti presenti. Selezionando uno di questi verranno visualizzate le informazioni e l'amministratore potrà inserire il numero di capi da aggiungere.

```
PRODOTTI PRESENTI NEL CATALOGO
1) Maglietta Blu
2) Jeans Corto
3) Giubbotto Nero
4) Pantalone Rosa
5) Cappello Nero
6) Costume Mimetico
7) Camicia Bianca

Selezionare il prodotto di cui modificare le scorte: 1

INFORMAZIONI PRODOTTO
Nome: Maglietta Blu
Prezzo: 10.50 euro
Taglie S disponibili: 4
Taglie M disponibili: 7
Taglie L disponibili: 20

1) Modifica il numero di taglie S
2) Modifica il numero di taglie M
3) Modifica il numero di taglie L
4) Indietro
1

Inserire il numero di taglie S da AGGIUNGERE: 5

Modifica effettuata con successo!
```

1.3.2 Aggiungere un nuovo prodotto al catalogo

Selezionando 2) compariranno una serie di campi che l'amministratore dovrà compilare, dopodichè il nuovo prodotto sarà inserito e visibile a tutti.

```
AGGIUNGI PRODOTTO AL CATALOGO
1) Aggiungi nuovo prodotto
2) Annulla
1

Nome Prodotto: Felpa
Caratteristica (es. Corto,Verde,Largo ecc.): Rossa
Prezzo in euro (es. 12.99): 24.99
Numero di taglie S disponibili: 3
Numero di taglie M disponibili: 6
Numero di taglie L disponibili: 12

Prodotto aggiunto correttamente
```

L'admin ha inserito una Felpa Rossa a 24.99 euro

2 Scelte Implementative

2.1 Strutture Dati Utilizzate

Come strutture dati ho utilizzato principalmente liste semplici, in eccezione della lista di attesa per la quale ho utilizzato una coda implementata tramite una lista doppiamente puntata.

Ho utilizzato le liste rispetto ad esempio agli array, dato che non mi serviva a molto avere l'accesso diretto agli elementi mentre era fondamentale poter modificare la struttura in qualunque momento (eliminazione e aggiunta di nodi ecc..).

In particolare ho utilizzato:

- Una **listaUtenti** per memorizzare tutti gli utenti
 - Ogni utente è una struct: nomeUtente, password, saldo
- Una **listaAmministratori** contenente tutti gli amministratori
- Una **listaProdotti** contenente tutti i prodotti
 - Ogni Prodotto è una struct: nomeProdotto, caratteristica, prezzo, taglieSdisponibili, taglieMdisponibili, taglieLdisponibili
- Un **carrello** che contiene i prodotti presenti nel carrello
 - Ogni prodotto nel carrello è una struct: nomeProdotto, caratteristica, tagliaRichiesta, prezzo
- Una **listaDiAttesa** che contiene i prodotti per cui gli utenti sono in attesa della disponibilità
 - E' una coda implementata con una lista doppiamente puntata, quindi ha un front ed un rear.
 - Ogni nodo ha quindi un campo next ed uno prev, serviranno per scorrere facilmente la coda in entrambe le direzioni.
 - Ogni elemento è una struct : nomeUtente, nomeProdotto, caratteristica, taglia

Tutte queste liste vengono popolate dai rispettivi file (utenti.txt prodotti.txt amministratori.txt e listaDiAttesa.txt) all'avvio del programma e saranno utilizzate dalle funzioni che gestiscono le varie funzionalità del programma.

2.2 Funzionalità Principali

Nel file funzioniShopping.c sono presenti tutte le funzioni definite nel programma, sembrano molte linee di codice ma in realtà gran parte sono occupate dalle varie printf scanf ecc..

Comunque alcune funzioni sono molto più complesse di altre quindi le spiego nel dettaglio per maggiore chiarezza.

2.2.1 Funzione latoUtente()

E' la funzione che gestisce tutto il lato utente dell'applicazione e viene chiamata proprio dal main quando l'utente sceglie di voler accedere al lato utente.

Prende in input l'utenteLoggato e tutte le liste, eccetto la listaAmministratori, come **doppio puntatore**. In questo modo sono riuscito a creare un'unica funzione che va a modificare direttamente il contenuto delle liste definite nel main. Senza doppi puntatori avrei dovuto realizzare piu' funzioni ognuna per ritornare una lista diversa.

Sostanzialmente la funzione chiede all'utente quale operazione effettuare ed in base alla scelta richiama delle altre funzioni più piccole come modificaSaldoNellaLista() o acquistaCarrello().

Richiama anche le funzioni gestisciAcquisto() e gestisciListaDiAttesa() che sono altrettanto complesse, infatti le spiegherò in seguito.

Si occupa anche di rendere persistenti le modifiche effettuate alle liste, infatti alla fine di ogni iterazione (quindi in pratica ogni volta che l'utente ritorna alla home dopo aver fatto un'operazione) va a sovrascrivere il contenuto dei file con il nuovo contenuto delle liste.

2.2.2 Funzioni gestisciAcquisto()

E' la funzione che si occupa di verificare che tutte le condizioni per effettuare un acquisto siano rispettate, ed agisce di conseguenza.

Segue sostanzialmente quello richiesto nella traccia del progetto, infatti:

- Se l'**utente ha abbastanza soldi e ci sono abbastanza taglie disponibili** allora l'acquisto può essere completato, quindi decrementa il saldo dell'utente, decrementa il numero di taglie disponibili del prodotto ed aggiorna le informazioni anche nelle rispettive liste.
- Se l'**utente non ha abbastanza soldi** allora fa semplicemente un inserimento in coda nella lista carrello, sempre sfruttando i doppi puntatori. Successivamente ritorna alla home per consentire all'utente di effettuare una ricarica.
- Se invece **il prodotto di quella taglia non è disponibile** fa un inserimento nella lista di attesa, che verrà fatto in testa dato che appunto la lista di attesa è una struttura di tipo FIFO.

2.2.3 Funzione gestisciListaDiAttesa()

E' probabilmente la funzione più complicata dell'intero progetto, infatti si occupa di capire quando un'utente ha effettivamente la priorità su un prodotto. Nel caso lo notifica e consente di aggiungere il prodotto al carrello.

Ricapitolando: listaDiAttesa è una lista doppiamente puntata quindi ogni suo nodo avrà un next ed un prev. Inoltre viene gestita come una coda quindi ha un front in cui inseriamo i nodi ed un rear da cui rimuoviamo i nodi.

Per prima cosa, dato che ho utilizzato un unica lista di attesa per tutti i prodotti, deve capire effettivamente utenteLoggato a quali prodotti è interessato.

Per fare questo si scorre tutta la lista di attesa ed ogni volta che incontra il nomeUtente dell'utenteLoggato va ad inserire il prodotto corrispondente in un array.

A fine iterazione avrò quindi un array prodottiInteressati[] che conterrà tutti i prodotti che si trovano nella lista di attesa dell'utenteLoggato.

Successivamente per ogni elemento dell'array, si scorre AL CONTRARIO la coda, quindi partendo dalla rear, ed il primo prodotto con lo stesso nome dell'elemento dell'array che incontrerà sarà quello con maggiore priorità. A questo punto se nel campo nomeUtente di questo elemento trovo il nomeUtente dell'utenteLoggato vorrà dire che ha effettivamente la priorità.

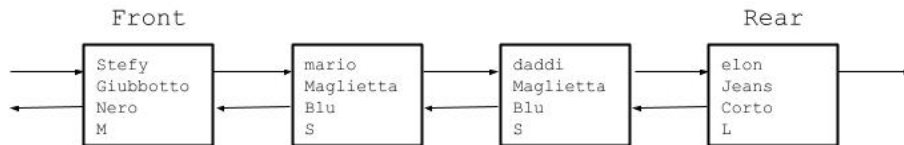
Esempio:

Nome utente loggato : daddi

Prodotto di cui verificare la priorità: Maglietta Blu S

Inizia a scorrere la lista da dietro fin quando non trova il primo "Maglietta Blu S", questo sarà il prodotto "Maglietta Blu S" con maggiore priorità.

A questo punto controlla il campo nomeUtente di questo elemento, se ci trova "daddi" vuol dire che utenteLoggato ha la priorità, altrimenti se ci trova ad esempio "mario" vuol dire che l'utente mario ha la priorità e quindi verrà notificato prima di daddi.



Qui daddi ha la priorità su Maglietta Blu S dato che è il primo iniziando a scorrere dalla rear, quindi verrà notificato per primo.

A questo punto se l'utente ha effettivamente la priorità va controllare la disponibilità del prodotto nella listaProdotti e se è maggiore di 0 (Quindi vuol dire che è tornato disponibile) notificherà l'utente.

L'utente potrà poi decidere se aggiungere questo prodotto al carrello oppure non fare niente, in entrambi i casi, dopo aver scelto, verrà RIMOSSO dalla lista di attesa per dare spazio a quelli dopo di lui per priorità.

2.3 Conclusioni

Sono presenti ovviamente molte altre funzioni ma dovrebbero essere abbastanza comprensibili dal codice commentato.

Molte di queste sono inoltre funzioni che abbiamo già visto nel corso di LASD ma riadattate ovviamente in base al contenuto delle liste.