

AmaTonLike

1.0

Generato da Doxygen 1.8.16

1 Indice delle strutture dati	1
1.1 Strutture dati	1
2 Indice dei file	3
2.1 Elenco dei file	3
3 Documentazione delle classi	5
3.1 Riferimenti per la struct acquirente_t	5
3.1.1 Descrizione dettagliata	5
3.1.2 Documentazione dei campi	5
3.1.2.1 cognome_acquirente	5
3.1.2.2 indirizzo_acquirente	6
3.1.2.3 nome_acquirente	6
3.1.2.4 psw	6
3.1.2.5 tel	6
3.1.2.6 tipo_acc	6
3.1.2.7 user	6
3.2 Riferimenti per la struct corriere_t	6
3.2.1 Descrizione dettagliata	7
3.2.2 Documentazione dei campi	7
3.2.2.1 cognome	7
3.2.2.2 nome	7
3.2.2.3 nome_azienza	7
3.2.2.4 psw	7
3.2.2.5 tel	7
3.2.2.6 tipo_acc	8
3.2.2.7 user	8
3.3 Riferimenti per la struct esercente_t	8
3.3.1 Descrizione dettagliata	8
3.3.2 Documentazione dei campi	8
3.3.2.1 cod_azienza	8
3.3.2.2 indirizzo_sede	9
3.3.2.3 nome_azienza	9
3.3.2.4 psw	9
3.3.2.5 tel	9
3.3.2.6 tipo_acc	9
3.3.2.7 user	9
3.4 Riferimenti per la struct prodotto_t	9
3.4.1 Descrizione dettagliata	10
3.4.2 Documentazione dei campi	10
3.4.2.1 acquirente	10
3.4.2.2 categoria	10
3.4.2.3 costo	10

3.4.2.4 esercente	10
3.4.2.5 id_ordine	11
3.4.2.6 id_prod	11
3.4.2.7 nome_prodotto	11
3.4.2.8 prodotti_venduti	11
3.4.2.9 quantita	11
3.4.2.10 spedizione	11
3.4.2.11 tot_prod_disp	11
4 Documentazione dei file	13
4.1 Riferimenti per il file src/Lib/constants.h	13
4.1.1 Descrizione dettagliata	13
4.1.2 Documentazione delle definizioni	13
4.1.2.1 LUNGHEZZA_PSW	14
4.1.2.2 LUNGHEZZA_TEL	14
4.1.2.3 STR	14
4.1.3 Documentazione delle variabili	14
4.1.3.1 acquirenti	14
4.1.3.2 acquisti	14
4.1.3.3 catalogo	14
4.1.3.4 corrieri	14
4.1.3.5 esercenti	15
4.2 Riferimenti per il file src/Lib/home_account.h	15
4.2.1 Descrizione dettagliata	15
4.2.2 Documentazione delle funzioni	15
4.2.2.1 home_acquirente()	15
4.2.2.2 home_corriere()	15
4.2.2.3 home_esercente()	16
4.3 Riferimenti per il file src/Lib/input.h	16
4.3.1 Descrizione dettagliata	16
4.3.2 Documentazione delle funzioni	16
4.3.2.1 controllo_num_string()	16
4.3.2.2 controllo_username()	17
4.3.2.3 incrementa_id_ordine()	17
4.3.2.4 incrementa_id_prodotto()	17
4.3.2.5 input_float()	18
4.3.2.6 input_int()	18
4.3.2.7 input_pass()	18
4.3.2.8 input_str()	18
4.3.2.9 input_tel()	19
4.4 Riferimenti per il file src/Lib/login.h	19
4.4.1 Descrizione dettagliata	19

4.4.2 Documentazione delle funzioni	19
4.4.2.1 accesso()	19
4.4.2.2 apri_files()	20
4.4.2.3 chiudi_files()	20
4.4.2.4 login()	20
4.4.2.5 registrazione()	20
4.4.2.6 tipo_account()	20
4.5 Riferimenti per il file src/Lib/menu_design.h	21
4.5.1 Descrizione dettagliata	21
4.5.2 Documentazione delle funzioni	21
4.5.2.1 menu_account_registrazione()	21
4.5.2.2 menu_acquirente()	22
4.5.2.3 menu_acquisto()	22
4.5.2.4 menu_categorie()	22
4.5.2.5 menu_corriere()	22
4.5.2.6 menu_esercente()	22
4.5.2.7 menu_filtro()	22
4.5.2.8 menu_login()	22
4.5.2.9 menu_ricerca_prodotto()	22
4.6 Riferimenti per il file src/Lib/specifiche.h	23
4.6.1 Descrizione dettagliata	23
4.6.2 Documentazione delle funzioni	23
4.6.2.1 acquisto()	23
4.6.2.2 cancellazione_acquisti()	24
4.6.2.3 modifica_stato_spedizione()	24
4.6.2.4 ordinamento_insert_sort()	24
4.6.2.5 ricerca_per_categoria()	24
4.6.2.6 ricerca_per_nome()	25
4.6.2.7 vendi_prodotto()	25
4.6.2.8 visualizza_acquisti()	25
4.6.2.9 visualizza_prodotti_esercente()	25
4.6.2.10 visualizza_spedizione()	26
4.7 Riferimenti per il file src/Lib/struct.h	26
4.7.1 Descrizione dettagliata	26
Indice analitico	27

Capitolo 1

Indice delle strutture dati

1.1 Strutture dati

Queste sono le strutture dati con una loro breve descrizione:

acquirente_t	5
corriere_t	6
esercente_t	8
prodotto_t	9

Capitolo 2

Indice dei file

2.1 Elenco dei file

Questo è un elenco dei file documentati con una loro breve descrizione:

src/Lib/	constants.h	13
src/Lib/	home_account.h	15
src/Lib/	input.h	16
src/Lib/	librerie.h	??
src/Lib/	login.h	19
src/Lib/	menu_design.h	21
src/Lib/	specifiche.h	23
src/Lib/	struct.h	26

Capitolo 3

Documentazione delle classi

3.1 Riferimenti per la struct acquirente_t

```
#include <struct.h>
```

Campi

- short int `tipo_acc`
- char `nome_acquirente` [STR]
- char `cognome_acquirente` [STR]
- char `indirizzo_acquirente` [STR]
- char `user` [STR]
- char `psw` [LUNGHEZZA_PSW]
- char `tel` [LUNGHEZZA_TEL]

3.1.1 Descrizione dettagliata

Struttura che contiene i dati utili per gli acquirenti

3.1.2 Documentazione dei campi

3.1.2.1 cognome_acquirente

```
char acquirente_t::cognome_acquirente[STR]
```

Cognome acquirente

3.1.2.2 indirizzo_acquirente

```
char acquirente_t::indirizzo_acquirente[STR]
```

Indirizzo acquirente

3.1.2.3 nome_acquirente

```
char acquirente_t::nome_acquirente[STR]
```

Nome acquirente

3.1.2.4 psw

```
char acquirente_t::psw[LUNGHEZZA_PSW]
```

Password

3.1.2.5 tel

```
char acquirente_t::tel[LUNGHEZZA_TEL]
```

Numero di telefono

3.1.2.6 tipo_acc

```
short int acquirente_t::tipo_acc
```

Tipo account

3.1.2.7 user

```
char acquirente_t::user[STR]
```

Username dell'acquirente

La documentazione per questa struct è stata generata a partire dal seguente file:

- [src/Lib/struct.h](#)

3.2 Riferimenti per la struct `corriere_t`

```
#include <struct.h>
```

Campi

- short int `tipo_acc`
- char `nome_azienza` [STR]
- char `nome` [STR]
- char `cognome` [STR]
- char `user` [STR]
- char `psw` [LUNGHEZZA_PSW]
- char `tel` [LUNGHEZZA_TEL]

3.2.1 Descrizione dettagliata

Struttura che contiene i dati utili per i corrieri

3.2.2 Documentazione dei campi

3.2.2.1 cognome

```
char corriere_t::cognome[STR]
```

Cognome del corriere

3.2.2.2 nome

```
char corriere_t::nome[STR]
```

Nome del corriere

3.2.2.3 nome_azienza

```
char corriere_t::nome_azienza[STR]
```

Nome dell'azienda del corriere

3.2.2.4 psw

```
char corriere_t::psw[LUNGHEZZA_PSW]
```

Password del corriere

3.2.2.5 tel

```
char corriere_t::tel[LUNGHEZZA_TEL]
```

Telefono del corriere

3.2.2.6 tipo_acc

```
short int corriere_t::tipo_acc
```

Tipo account

3.2.2.7 user

```
char corriere_t::user[STR]
```

Username del corriere

La documentazione per questa struct è stata generata a partire dal seguente file:

- [src/Lib/struct.h](#)

3.3 Riferimenti per la struct esercente_t

```
#include <struct.h>
```

Campi

- short int [tipo_acc](#)
- char [nome_azienza](#) [STR]
- char [cod_azienza](#) [STR]
- char [indirizzo_sede](#) [STR]
- char [user](#) [STR]
- char [psw](#) [LUNGHEZZA_PSW]
- char [tel](#) [LUNGHEZZA_TEL]

3.3.1 Descrizione dettagliata

Struttura che contiene i dati utili per gli esercenti

3.3.2 Documentazione dei campi

3.3.2.1 cod_azienza

```
char esercente_t::cod_azienza[STR]
```

Codice azienda

3.3.2.2 indirizzo_sede

```
char esercente_t::indirizzo_sede[STR]
```

Indirizzo dell'azienda

3.3.2.3 nome_azienza

```
char esercente_t::nome_azienza[STR]
```

Nome azienda venditrice

3.3.2.4 psw

```
char esercente_t::psw[LUNGHEZZA_PSW]
```

Password dell'esercente

3.3.2.5 tel

```
char esercente_t::tel[LUNGHEZZA_TEL]
```

Numero di telefono dell'esercente

3.3.2.6 tipo_acc

```
short int esercente_t::tipo_acc
```

Tipo account

3.3.2.7 user

```
char esercente_t::user[STR]
```

Username dell'esercente

La documentazione per questa struct è stata generata a partire dal seguente file:

- src/Lib/[struct.h](#)

3.4 Riferimenti per la struct prodotto_t

```
#include <struct.h>
```

Campi

- int `id_ordine`
- int `id_prod`
- int `tot_prod_disp`
- int `prodotti_venduti`
- int `quantita`
- float `costo`
- char `nome_prodotto` [STR]
- char `categoria` [STR]
- char `spedizione` [STR]
- char `esercente` [STR]
- char `acquirente` [STR]

3.4.1 Descrizione dettagliata

Struttura che contiene i dati del prodotto

3.4.2 Documentazione dei campi

3.4.2.1 acquirente

```
char prodotto_t::acquirente[STR]
```

Acquirente prodotto

3.4.2.2 categoria

```
char prodotto_t::categoria[STR]
```

Categoria del prodotto

3.4.2.3 costo

```
float prodotto_t::costo
```

Costo del prodotto

3.4.2.4 esercente

```
char prodotto_t::esercente[STR]
```

Venditore prodotto

3.4.2.5 `id_ordine`

```
int prodotto_t::id_ordine
```

ID dell'ordine del prodotto

3.4.2.6 `id_prod`

```
int prodotto_t::id_prod
```

ID del prodotto

3.4.2.7 `nome_prodotto`

```
char prodotto_t::nome_prodotto[STR]
```

Nome del prodotto

3.4.2.8 `prodotti_venduti`

```
int prodotto_t::prodotti_venduti
```

Numero dei prodotti venduti

3.4.2.9 `quantita`

```
int prodotto_t::quantita
```

La quantità ha due significati: per l'acquirente quanti prodotti vuole acquistare e per l'esercente quanti ne vuole mettere inv vendita

3.4.2.10 `spedizione`

```
char prodotto_t::spedizione[STR]
```

Stato spedizione del prodotto

3.4.2.11 `tot_prod_disp`

```
int prodotto_t::tot_prod_disp
```

Totale dei prodotti disponibili

La documentazione per questa struct Ã¨ stata generata a partire dal seguente file:

- `src/Lib/struct.h`

Capitolo 4

Documentazione dei file

4.1 Riferimenti per il file src/Lib/constants.h

```
#include "librerie.h"
```

Definizioni

- #define STR 50
- #define LUNGHEZZA_PSW 5
- #define LUNGHEZZA_TEL 10

Variabili

- FILE * acquirenti
- FILE * esercenti
- FILE * corrieri
- FILE * catalogo
- FILE * acquisti

4.1.1 Descrizione dettagliata

Created on: 9 giu 2021

Libreria che contiene le costanti utili al programma

Autori

: Alessandro, Alessia

4.1.2 Documentazione delle definizioni

4.1.2.1 LUNGHEZZA_PSW

```
#define LUNGHEZZA_PSW 5
```

Lunghezza massima per la password

4.1.2.2 LUNGHEZZA_TEL

```
#define LUNGHEZZA_TEL 10
```

Lunghezza massima per il numero di telefono

4.1.2.3 STR

```
#define STR 50
```

Lunghezza per le stringhe generali

4.1.3 Documentazione delle variabili

4.1.3.1 acquirenti

```
FILE* acquirenti
```

Variabile di tipo FILE per gli acquirenti

4.1.3.2 acquisti

```
FILE* acquisti
```

Variabili di tipo FILE per gli acquisti

4.1.3.3 catalogo

```
FILE* catalogo
```

Variabili di tipo FILE per il catalogo

4.1.3.4 corrieri

```
FILE* corrieri
```

Variabili di tipo FILE per i corrieri

4.1.3.5 esercenti

FILE* esercenti

Variabili di tipo FILE per gli esercenti

4.2 Riferimenti per il file src/Lib/home_account.h

Funzioni

- void [home_acquirente](#) (char *user)
- void [home_esercente](#) (char *user)
- void [home_corriere](#) ()

4.2.1 Descrizione dettagliata

Created on: 16 giu 2021

Libreria che contiene le procedure utili per i menu'/home degli utenti

Autori

: Alessandro, Alessia

4.2.2 Documentazione delle funzioni

4.2.2.1 home_acquirente()

```
void home_acquirente (
    char * user )
```

Procedura che si occupa delle funzioni che l'acquirente puo' eseguire per gli acquisti e per la visualizzazione di essi in base all'utente che ha effettuato l'accesso.

Parametri

<i>user</i>	dell'utente che ha effettuato l'accesso
-------------	---

4.2.2.2 home_corriere()

```
void home_corriere ( )
```

Procedura che si occupa delle funzioni che il corriere puo' eseguire.

4.2.2.3 home_esercente()

```
void home_esercente (
    char * user )
```

Procedura che si occupa delle funzioni che l'esercente puo' eseguire per la vendita e per la visualizzazione degli articoli messi in vendita in base all'utente che ha effettuato l'accesso.

Parametri

<i>user</i>	dell'utente che ha effettuato l'accesso
-------------	---

4.3 Riferimenti per il file src/Lib/input.h

Funzioni

- int [input_int](#) ()
- float [input_float](#) ()
- char * [input_str](#) ()
- char * [input_tel](#) ()
- char * [input_pass](#) ()
- int [controllo_num_string](#) (char *stringa)
- int [controllo_username](#) (char *input)
- int [incrementa_id_prodotto](#) ()
- int [incrementa_id_ordine](#) ()

4.3.1 Descrizione dettagliata

Created on: 9 giu 2021

Libreria che si occupa degli inserimenti in input e dei controlli degli inserimenti e l'incremento degli id per quanto riguarda i prodotti

Autori

: Alessandro, Alessia

4.3.2 Documentazione delle funzioni

4.3.2.1 controllo_num_string()

```
int controllo_num_string (
    char * stringa )
```

Funzione che controlla se la stringa presa come parametro contiene numeri.

Parametri

<i>stringa</i>	da controllare.
----------------	-----------------

Restituisce

1 nel caso ci sono numeri o 0 se non ci sono.

4.3.2.2 controllo_username()

```
int controllo_username (
    char * input )
```

Funzione che legge nei file utente, se trova l'username preso come parametro, all'interno di un file, allora l'username e' gia' esistente.

Parametri

<i>input</i>	inserito dall'utente.
--------------	-----------------------

Restituisce

0 se l'username non esista gia', 1 se l'username esiste gia'.

4.3.2.3 incrementa_id_ordine()

```
int incrementa_id_ordine ( )
```

Funzione che legge nel file acquisti l'ultimo prodotto controllandone l'id ordine ed incrementandolo di uno

Restituisce

l'id incrementato.

4.3.2.4 incrementa_id_prodotto()

```
int incrementa_id_prodotto ( )
```

Funzione che legge nel file catalogo l'ultimo prodotto controllandone l'id prodotto ed incrementandolo di uno.

Restituisce

l'id incrementato.

4.3.2.5 input_float()

```
float input_float ( )
```

Funzione che si occupa dell'inserimento di float con programmazione difensiva nel caso non viene inserito un float.

Restituisce

il numero decimale inserito.

4.3.2.6 input_int()

```
int input_int ( )
```

Funzione che si occupa dell'inserimento di interi con programmazione difensiva nel caso non viene inserito un intero.

Restituisce

l'intero inserito.

4.3.2.7 input_pass()

```
char* input_pass ( )
```

Funzione che si occupa dell'inserimento della password con programmazione difensiva nel caso la password e' piu' lunga di 5 caratteri e non contiene una maiuscola ed un numero.

Restituisce

la password

4.3.2.8 input_str()

```
char* input_str ( )
```

Funzione che si occupa dell'inserimento di stringhe con programmazione difensiva nel caso non viene inserita una stringa o la stringa inserita e' vuota.

Restituisce

la stringa inserita.

4.3.2.9 input_tel()

```
char* input_tel ( )
```

Funzione che si occupa dell'inserimento del numero di telefono con programmazione difensiva nel caso il numero inserito non e' lungo 10 numeri o nel caso non ci sono numeri.

Restituisce

il numero di telefono.

4.4 Riferimenti per il file src/Lib/login.h

Funzioni

- char * [login](#) ()
- char * [accesso](#) ()
- void [registrazione](#) ()
- int [tipo_account](#) (char *user)
- int [apri_files](#) ()
- void [chiudi_files](#) ()

4.4.1 Descrizione dettagliata

Created on: 9 giu 2021

Libreria che contiene le funzioni e procedure utili all' inizio del programma e alla fase di login

Autori

: Alessandro, Alessia

4.4.2 Documentazione delle funzioni

4.4.2.1 accesso()

```
char* accesso ( )
```

Funzione che si occupa dell'accesso al programma, chiedendo in input all'utente username e password che verranno confrontati con quelli registrati.

Restituisce

l'username dell'utente che ha effettuato l'accesso.

4.4.2.2 apri_files()

```
int apri_files ( )
```

Funzione che apre tutti i file

Restituisce

1 se ha successo l'apertura, 0 altrimenti

4.4.2.3 chiudi_files()

```
void chiudi_files ( )
```

Procedura che chiude tutti i file

4.4.2.4 login()

```
char* login ( )
```

Funzione che si occupa della fase di login e permette all'utente di scegliere se accedere o registra. Nel caso l'utente scelga di accedere, il programma memorizzerà l'username con cui si è effettuato l'accesso. Nel caso l'utente scelga di registrarsi, il programma acquisirà le informazioni utili alla registrazione e tornerà alla fase login.

Restituisce

l'username dell'utente che ha effettuato l'accesso.

4.4.2.5 registrazione()

```
void registrazione ( )
```

Procedura che acquisisce i dati utili per la registrazione di un nuovo account nel programma. L'utente dovrà prima scegliere che tipo di account registrare ed in base al tipo scelto, verranno chiesti determinati dati.

4.4.2.6 tipo_account()

```
int tipo_account (
    char * user )
```

Funzione che prende come parametro l'username con cui si è effettuato l'accesso, ne ricava il tipo ritornandolo, controllando in quale file si trova quell'username.

Parametri

<i>user</i>	dell'utente che ha effettuato l'accesso.
-------------	--

Restituisce

1 se trova l'username nel file acquirenti, 2 se trova l'username nel file esercenti, 3 se trova l'username nel file corrieri.

4.5 Riferimenti per il file src/Lib/menu_design.h

```
#include "librerie.h"
```

Funzioni

- void [menu_login](#) ()
- void [menu_account_registrazione](#) ()
- void [menu_acquirente](#) ()
- void [menu_esercente](#) ()
- void [menu_corriere](#) ()
- void [menu_categorie](#) ()
- void [menu_ricerca_prodotto](#) ()
- void [menu_acquisto](#) ()
- void [menu_filtro](#) ()

4.5.1 Descrizione dettagliata

Created on: 9 giu 2021

Libreria che contiene le implementazioni delle procedure che gestiscono le interfacce grafiche del programma

Autori

: Alessandro, Alessia

4.5.2 Documentazione delle funzioni

4.5.2.1 [menu_account_registrazione\(\)](#)

```
void menu_account_registrazione ( )
```

Menu registrazione

4.5.2.2 menu_acquirente()

```
void menu_acquirente ( )
```

Menu acquirente

4.5.2.3 menu_acquisto()

```
void menu_acquisto ( )
```

Menu acquisto

4.5.2.4 menu_categorie()

```
void menu_categorie ( )
```

Menu categorie

4.5.2.5 menu_corriere()

```
void menu_corriere ( )
```

Menu corriere

4.5.2.6 menu_esercente()

```
void menu_esercente ( )
```

Menu esercente

4.5.2.7 menu_filtro()

```
void menu_filtro ( )
```

Menu filtri

4.5.2.8 menu_login()

```
void menu_login ( )
```

Menu login

4.5.2.9 menu_ricerca_prodotto()

```
void menu_ricerca_prodotto ( )
```

Menu di ricerca del prodotto

4.6 Riferimenti per il file src/Lib/specifiche.h

```
#include "struct.h"
```

Funzioni

- int [ricerca_per_categoria](#) ()
- int [ricerca_per_nome](#) ()
- void [ordinamento_insert_sort](#) (prodotto_t *array, int dim, int scelta)
- void [acquisto](#) (char *user)
- void [cancellazione_acquisti](#) (char *user)
- void [visualizza_acquisti](#) (char *user)
- void [vendi_prodotto](#) (char *user)
- void [visualizza_prodotti_esercente](#) (char *user)
- void [visualizza_spedizione](#) ()
- void [modifica_stato_spedizione](#) ()

4.6.1 Descrizione dettagliata

Created on: 16 giu 2021

Libreria che contiene tutte le funzioni e procedure che riguardano le azioni che gli utenti possono fare nel programma.

Autori

: Alessandro, Alessia

4.6.2 Documentazione delle funzioni

4.6.2.1 acquisto()

```
void acquisto (  
    char * user )
```

Procedura che permette all'acquirente di acquistare un prodotto che ha precedentemente cercato.

Parametri

<i>user</i>	(username) dell'utente che ha effettuato l'accesso
-------------	--

4.6.2.2 cancellazione_acquisti()

```
void cancellazione_acquisti (
    char * user )
```

Procedura che permette all'acquirente di cancellare un ordine non ancora consegnato.

Parametri

<i>user</i>	(username) dell'utente che ha effettuato l'accesso
-------------	--

4.6.2.3 modifica_stato_spedizione()

```
void modifica_stato_spedizione ( )
```

Procedura che permette al corriere di modificare lo stato di spedizione di un prodotto. La modifica avviene da "in attesa" a "spedito", da "spedito" a "consegnato".

4.6.2.4 ordinamento_insert_sort()

```
void ordinamento_insert_sort (
    prodotto_t * array,
    int dim,
    int scelta )
```

Procedura che si occupa dell'ordinamento e della stampa dei prodotti che sono stati precedentemente cercati tramite le funzioni di ricerca.

Parametri

<i>array</i>	dei prodotti da ordinare
<i>dim</i>	(dimensione) dell'array
<i>scelta</i>	del filtro, se si vuole ordinare per costo crescente, costo decrescente o non si vuole ordinare

4.6.2.5 ricerca_per_categoria()

```
int ricerca_per_categoria ( )
```

Funzione che si occupa della ricerca dei prodotti in base alla categoria scelta dall'utente. Il risultato della ricerca viene passato ad un array che successivamente viene ordinato (se l'utente sceglie un filtro per l'ordinamento del risultato).

Restituisce

1 se la ricerca non ha trovato prodotti nel file catalogo, oppure 0 se ha trovato dei prodotti

4.6.2.6 ricerca_per_nome()

```
int ricerca_per_nome ( )
```

Funzione che si occupa della ricerca dei prodotti in base al nome del prodotto (la ricerca puo' avvenire anche con sottostringa). Il risultato della ricerca viene passato ad un array che successivamente viene ordinato (se l'utente sceglie un filtro per l'ordinamento del risultato).

Restituisce

1 se la ricerca non ha prodotto risultato, oppure 0 se la ricerca e' andata a buon fine.

4.6.2.7 vendi_prodotto()

```
void vendi_prodotto (
    char * user )
```

Procedura che permette all'esercente di inserire nel catalogo un prodotto, inserendo i dati richiesti.

Parametri

<i>user</i>	(username) dell'utente che ha effettuato l'accesso
-------------	--

4.6.2.8 visualizza_acquisti()

```
void visualizza_acquisti (
    char * user )
```

Procedura che permette all'acquirente di visualizzare tutti gli acquisti effettuati.

Parametri

<i>user</i>	(username) dell'utente che ha effettuato l'accesso
-------------	--

4.6.2.9 visualizza_prodotti_esercente()

```
void visualizza_prodotti_esercente (
    char * user )
```

Procedura che permette all'esercente di visualizzare i suoi prodotti in vendita e di visualizzarne le statistiche come prodotto con piu' vendite, prodotto piu' costoso messo in vendita, numero prodotti per categoria.

Parametri

<i>user</i>	(username) dell'utente che ha effettuato l'accesso
-------------	--

4.6.2.10 visualizza_spedizione()

```
void visualizza_spedizione ( )
```

Procedura che permette al corriere di visualizzare tutti gli stati di spedizione dei prodotti acquistati e di prenderne il carico. Non può visualizzare i prodotti già consegnati.

4.7 Riferimenti per il file src/Lib/struct.h

```
#include "constants.h"
```

Strutture dati

- struct [acquirente_t](#)
- struct [esercente_t](#)
- struct [corriere_t](#)
- struct [prodotto_t](#)

4.7.1 Descrizione dettagliata

Created on: 9 giu 2021

Libreria che contiene le struct che gestiscono i dati degli utenti e dei prodotti

Autori

: Alessandro, Alessia

Indice analitico

accesso
 login.h, 19

acquirente
 prodotto_t, 10

acquirente_t, 5
 cognome_acquirente, 5
 indirizzo_acquirente, 5
 nome_acquirente, 6
 psw, 6
 tel, 6
 tipo_acc, 6
 user, 6

acquirenti
 constants.h, 14

acquisti
 constants.h, 14

acquisto
 specifiche.h, 23

apri_files
 login.h, 19

cancellazione_acquisti
 specifiche.h, 23

catalogo
 constants.h, 14

categoria
 prodotto_t, 10

chiudi_files
 login.h, 20

cod_azienza
 esercente_t, 8

cognome
 corriere_t, 7

cognome_acquirente
 acquirente_t, 5

constants.h
 acquirenti, 14
 acquisti, 14
 catalogo, 14
 corrieri, 14
 esercenti, 14
 LUNGHEZZA_PSW, 13
 LUNGHEZZA_TEL, 14
 STR, 14

controllo_num_string
 input.h, 16

controllo_username
 input.h, 17

corriere_t, 6
 cognome, 7
 nome, 7
 nome_azienza, 7
 psw, 7
 tel, 7
 tipo_acc, 7
 user, 8

corrieri
 constants.h, 14

costo
 prodotto_t, 10

esercente
 prodotto_t, 10

esercente_t, 8
 cod_azienza, 8
 indirizzo_sede, 8
 nome_azienza, 9
 psw, 9
 tel, 9
 tipo_acc, 9
 user, 9

esercenti
 constants.h, 14

home_account.h
 home_acquirente, 15
 home_corriere, 15
 home_esercente, 15

home_acquirente
 home_account.h, 15

home_corriere
 home_account.h, 15

home_esercente
 home_account.h, 15

id_ordine
 prodotto_t, 10

id_prod
 prodotto_t, 11

incrementa_id_ordine
 input.h, 17

incrementa_id_prodotto
 input.h, 17

indirizzo_acquirente
 acquirente_t, 5

indirizzo_sede
 esercente_t, 8

input.h
 controllo_num_string, 16
 controllo_username, 17

- incrementa_id_ordine, [17](#)
- incrementa_id_prodotto, [17](#)
- input_float, [17](#)
- input_int, [18](#)
- input_pass, [18](#)
- input_str, [18](#)
- input_tel, [18](#)
- input_float
 - input.h, [17](#)
- input_int
 - input.h, [18](#)
- input_pass
 - input.h, [18](#)
- input_str
 - input.h, [18](#)
- input_tel
 - input.h, [18](#)
- login
 - login.h, [20](#)
- login.h
 - accesso, [19](#)
 - apri_files, [19](#)
 - chiudi_files, [20](#)
 - login, [20](#)
 - registrazione, [20](#)
 - tipo_account, [20](#)
- LUNGHEZZA_PSW
 - constants.h, [13](#)
- LUNGHEZZA_TEL
 - constants.h, [14](#)
- menu_account_registrazione
 - menu_design.h, [21](#)
- menu_acquirente
 - menu_design.h, [21](#)
- menu_acquisto
 - menu_design.h, [22](#)
- menu_categorie
 - menu_design.h, [22](#)
- menu_corriere
 - menu_design.h, [22](#)
- menu_design.h
 - menu_account_registrazione, [21](#)
 - menu_acquirente, [21](#)
 - menu_acquisto, [22](#)
 - menu_categorie, [22](#)
 - menu_corriere, [22](#)
 - menu_esercente, [22](#)
 - menu_filtro, [22](#)
 - menu_login, [22](#)
 - menu_ricerca_prodotto, [22](#)
- menu_esercente
 - menu_design.h, [22](#)
- menu_filtro
 - menu_design.h, [22](#)
- menu_login
 - menu_design.h, [22](#)
- menu_ricerca_prodotto
 - menu_design.h, [22](#)
- modifica_stato_spedizione
 - specifiche.h, [24](#)
- nome
 - corriere_t, [7](#)
- nome_acquirente
 - acquirente_t, [6](#)
- nome_azienza
 - corriere_t, [7](#)
 - esercente_t, [9](#)
- nome_prodotto
 - prodotto_t, [11](#)
- ordinamento_insert_sort
 - specifiche.h, [24](#)
- prodotti_venduti
 - prodotto_t, [11](#)
- prodotto_t, [9](#)
 - acquirente, [10](#)
 - categoria, [10](#)
 - costo, [10](#)
 - esercente, [10](#)
 - id_ordine, [10](#)
 - id_prod, [11](#)
 - nome_prodotto, [11](#)
 - prodotti_venduti, [11](#)
 - quantita, [11](#)
 - spedizione, [11](#)
 - tot_prod_disp, [11](#)
- psw
 - acquirente_t, [6](#)
 - corriere_t, [7](#)
 - esercente_t, [9](#)
- quantita
 - prodotto_t, [11](#)
- registrazione
 - login.h, [20](#)
- ricerca_per_categoria
 - specifiche.h, [24](#)
- ricerca_per_nome
 - specifiche.h, [24](#)
- specifiche.h
 - acquisto, [23](#)
 - cancellazione_acquisti, [23](#)
 - modifica_stato_spedizione, [24](#)
 - ordinamento_insert_sort, [24](#)
 - ricerca_per_categoria, [24](#)
 - ricerca_per_nome, [24](#)
 - vendi_prodotto, [25](#)
 - visualizza_acquisti, [25](#)
 - visualizza_prodotti_esercente, [25](#)
 - visualizza_spedizione, [26](#)
- spedizione
 - prodotto_t, [11](#)
- src/Lib/constants.h, [13](#)

src/Lib/home_account.h, [15](#)

src/Lib/input.h, [16](#)

src/Lib/login.h, [19](#)

src/Lib/menu_design.h, [21](#)

src/Lib/specifiche.h, [23](#)

src/Lib/struct.h, [26](#)

STR

constants.h, [14](#)

tel

acquirente_t, [6](#)

corriere_t, [7](#)

esercente_t, [9](#)

tipo_acc

acquirente_t, [6](#)

corriere_t, [7](#)

esercente_t, [9](#)

tipo_account

login.h, [20](#)

tot_prod_disp

prodotto_t, [11](#)

user

acquirente_t, [6](#)

corriere_t, [8](#)

esercente_t, [9](#)

vendi_prodotto

specifiche.h, [25](#)

visualizza_acquisti

specifiche.h, [25](#)

visualizza_prodotti_esercente

specifiche.h, [25](#)

visualizza_spedizione

specifiche.h, [26](#)