## 

Caso di Studio: Traccia 6

**Stabilimento balneare:**

**Por do Sol**

Laboratorio di Informatica

02/11/2018

A.A. 2018/19

Autore:

*Matteo Luceri*

*matricola #677663*

## **1.Analisi**

## 1.1 Descrizione del Sistema

Il sistema simula una piattaforma di gestione automatizzata di uno stabilimento balneare. Nello specifico, il programma dovrà occuparsi di ciò che concerne la gestione degli accessi al sistema (l’accesso è in relazione ai dati acquisiti da file), la modifica dei dati relativi agli utenti accedenti, la gestione delle prenotazioni giornaliere relative agli ombrelloni e la conseguente gestione sulle disponibilità di questi. Il sistema si occuperà inoltre della modifica nonché rimozione delle prenotazioni effettuatesi e sulla stampa come output delle prenotazioni per ogni data.

I **clienti** dal sistema saranno definiti attraverso le seguenti, obbligatorie, caratteristiche:

* E-mail (equivalente ad un nickname, univoco)
* Password
* Nome
* Cognome
* Data di nascita
* Codice fiscale
* Percentuale di sconto

Le **prenotazioni** dal sistema saranno definiti attraverso le seguenti caratteristiche:

* Codice progressivo di prenotazione (univoco)
* Data della prenotazione (ben formata)
* Codice Fiscale
* Tipo di prenotazione:
  + Mattina (su file/sistema : 1, a video : M)
  + Pomeriggio (su file/sistema : 2, a video : P)
  + Intera giornata (su file/sistema : 3 , a video : G)
  + Doppia prenotazione (su file : /,su sistema: 0 , a video : G)
* Il numero dell’ombrellone (max 35 per fila)
* Il numero di fila dell’ombrellone (max 10)
* Il numero di lettini (2 di default)
* Il numero di sedie (0 di default)
* Il costo dell’ombrellone così calcolato:
  + Per ombrellone con 2 lettini: 25€ per le file 1-2, 20€ per le file 3-4, 15€ per le restanti file;
  + Per ogni lettino aggiuntivo: 10€ per le file 1-2, 8€ per le file 3-4, 5€ per le restanti file;
  + Per ogni sedia aggiuntiva: 7€ per le file 1-2, 5€ per le file 3-4, 3€ per le restanti file.

(Se la prenotazione è di tipo M o P il costo si dimezza. Se il cliente ha uno sconto applicarlo alla fine.)

Le **relazioni cliente-prenotazione** dal sistema sarà definita attraversa l’indirizzo e-mail.

## 1.2 Requisiti Funzionali

|  |  |  |
| --- | --- | --- |
| **Codice** | **Nome** | **Descrizione** |
| **A00** | Caricamento/conteggio dei dati da file “clienti.csv” | Il programma carica:  i dati dei clienti, e conta le righe. |
| **A01** | Caricamento/conteggio dei dati da file “prenotazioni.csv” | Il programma carica:  i dati delle prenotazioni, e conta le righe. |
| **A02** | Gestione dell’accesso al sistema | Il programma deve verificare le credenziali di accesso di un cliente. |
| **A03** | Ricerca /confronto del campo email | Il programma deve ricercare l’e-mail data per input e ricercarla nel file “clienti.csv”. |
| **A04** | Ricerca /confronto del campo password | Il programma deve ricercare la password data per input e ricercarla nel file “clienti.csv”. |
| **B05** | Modifica dei dati di un cliente | Il programma deve modificare i dati relativi ad un cliente già registrato. |
| **B06** | Controllo di una data | Il programma deve verificare la validità di una data. |
| **B07** | Stampa su file la lista di clienti | Il programma deve stampare sul file “clienti.csv” la lista dei clienti. |
| **C08** | Acquisizione di una prenotazione | Il programma deve acquisire una nuova prenotazione. |
| **C09** | Ricerca una data | Il programma deve ricercare una data ricevuta per input. |
| **C10** | Ricerca numero di fila ed ombrellone | Il programma deve ricercare una numero di fila e di ombrellone ricevuto per input. |
| **C11** | Verifica la disponibilità di una ombrellone | Il programma deve verificare la disponibilità di un ombrellone. |
| **C12** | Acquisizione tipo di prenotazione | Il programma deve acquisire un tipo di prenotazione. |
| **C13** | Acquisizione numero di lettini | Il programma deve acquisire un numero di lettini. |
| **C14** | Acquisizione numero di sedie | Il programma deve acquisire un numero di sedie. |
| **C15** | Calcolo del costo totale | Il programma deve calcolare il costo totale di una prenotazione. |
| **C16** | Stampa a video del tipo di prenotazione | Il programma deve stampare a video il tipo di prenotazione. |
| **C17** | Stampa su file una nuova prenotazione | Il programma deve stampare su file una nuova prenotazione. |
| **D18** | Verifica la disponibilità di una fila di ombrelloni | Il programma deve verificare la disponibilità di una fila di ombrelloni. |
| **D19** | Ricerca numero di fila fra le prenotazioni | Il programma deve ricercare una numero di fila fra le prenotazioni. |
| **D20** | Elabora la disponibilità del giorno | Il programma deve calcolare la disponibilità di un giorno. |
| **D21** | Stampa a video la fila di ombrelloni disponibili | Il programma deve stampare a schermo la disponibilità di una fila di ombrelloni. |
| **E22** | Stampa a video le file di ombrelloni, in ordine crescente/decrescente, disponibili | Il programma deve stampare a schermo la disponibilità di tutti gli ombrelloni. |
| **E23** | Verifica la disponibilità di tutti gli ombrelloni | Il programma deve verificare la disponibilità di tutti gli ombrelloni. |
| **F24** | Modifica dati di una prenotazione | Il programma deve modificare i dati di una prenotazione. |
| **F25** | Stampa a video una prenotazione | Il programma deve stampare una prenotazione a video. |
| **F26** | Modifica tipo di prenotazione | Il programma deve modificare il tipo di prenotazione. |
| **F27** | Modifica il numero di ombrellone | Il programma deve modificare il numero di prenotazione. |
| **F28** | Aggiorna file prenotazioni | Il programma deve aggiornare il file “prenotazioni.csv”. |
| **G29** | Rimozione di una prenotazione | Il programma deve rimuovere una prenotazione scelta. |
| **G30** | Stampa su file le prenotazioni non rimosse | Il programma deve stampare le prenotazioni non rimosse dalla lista. |
| **H31** | Stampa su file tutte le prenotazioni per ogni data | Il programma deve stampare su file tutte le prenotazioni divise per ogni data. |
| **H32** | Ordinamento delle prenotazioni per data | Il programma deve ordinare tutte le le prenotazioni per data. |
| **H33** | Calcolo della data estesa | l programma deve calcolare il valore di una data per esteso. |

s

Note: Ad ogni funzione è stata associata una lettera. Ad ogni lettera è associabile un gruppo di funzioni, significativo.

Il numero di ogni funzione è progressivo

## 1.3 Casi d’Uso

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| A00 | Caricamento/conteggio dei dati da file “clienti.csv” | | Il programma carica  i dati dei clienti, e conta le righe. |
| Pre-Condizioni | I file devono essere validi:   * Estensione .csv * Nomi devono essere quelli stabiliti dal programmatore * Devono obbligatoriamente essere presenti tutti (anche se vuoti) * Le formattazioni stabilite per i contenuti dei file devono essere rispettate | | |
| Post-Condizioni | *Successo:* | Il programma carica con successo i dati presenti sui file e continua l’esecuzione | |
| *Fallimento:* | Il programma mostra un messaggio di errore sull’apertura dei file | |
| Evento innescante | Apertura del software o modifica di file a runtime che prevede un ripristino dei dati caricati dal programma | | |
| Scenario di base | Il funzionamento del programma è corretto e tutti i dati sono stati prelevati e sono utilizzabili nel resto del programma | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso e non viene prelevato nessun dato. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| A01 | Caricamento/conteggio dei dati da file “prenotazioni.csv” | | Il programma carica:  i dati delle prenotazioni, e conta le righe. |
| Pre-Condizioni | I file devono essere validi:   * Estensione .csv * Nomi devono essere quelli stabiliti dal programmatore * Devono obbligatoriamente essere presenti tutti (anche se vuoti) * Le formattazioni stabilite per i contenuti dei file devono essere rispettate | | |
| Post-Condizioni | *Successo:* | Il programma carica con successo i dati presenti sui file e continua l’esecuzione | |
| *Fallimento:* | Il programma mostra un messaggio di errore sull’apertura dei file | |
| Evento innescante | Apertura del software o modifica di file a runtime che prevede un ripristino dei dati caricati dal programma | | |
| Scenario di base | Il funzionamento del programma è corretto e tutti i dati sono stati prelevati e sono utilizzabili nel resto del programma | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso e non viene prelevato nessun dato. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| A02 | Gestione dell’accesso al sistema | | Il programma deve verificare le credenziali di accesso di un cliente. |
| Pre-Condizioni | I dati da file devono essere stati caricati correttamente. I dati di input devono essere validi. | | |
| Post-Condizioni | *Successo:* | Il programma permette l’accesso. | |
| *Fallimento:* | Il programma non permette l’accesso. | |
| Evento innescante | Apertura del software. | | |
| Scenario di base | Il funzionamento del programma è corretto qualora permetta un accesso/uscita dovutamente controllato. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso quando l’accesso o l’uscita non sono controllati. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| A03 | Ricerca /confronto del campo email | | Il programma deve ricercare l’e-mail data per input nel file “clienti.csv”. |
| Pre-Condizioni | I dati da file devono essere stati caricati correttamente. I dati di input devono essere validi. | | |
| Post-Condizioni | *Successo:* | Il programma confronta l’input con i dati, trovando l’e-mail. | |
| *Fallimento:* | Il programma presenta un messaggio di errore. | |
| Evento innescante | Inserimento delle credenziali di accesso. | | |
| Scenario di base | Il funzionamento del programma è corretto qualora acquisito l’input, la funzione ricerca ed effettua con successo il confronto. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso quando l’input inserito non è confrontato | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| A04 | Ricerca /confronto del campo password | | Il programma deve ricercare la password data per input e ricercarla nel file “clienti.csv”. |
| Pre-Condizioni | I dati da file devono essere stati caricati correttamente. I dati di input devono essere validi. | | |
| Post-Condizioni | *Successo:* | Il programma confronta l’input con i dati, trovando la password. | |
| *Fallimento:* | Il programma presenta un messaggio di errore. | |
| Evento innescante | Inserimento delle credenziali di accesso.  Confronto dell’email avvenuto con successo. | | |
| Scenario di base | Il funzionamento del programma è corretto qualora acquisito l’input, la funzione ricerca ed effettua con successo il confronto. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso quando l’input inserito non è confrontato | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| B05 | Modifica dei dati di un cliente | | Il programma deve modificare i dati relativi ad un cliente già registrato. |
| Pre-Condizioni | Aver effettuato l’accesso con successo. | | |
| Post-Condizioni | *Successo:* | Il sistema procede alla modifica dei dati del giocatore | |
| *Fallimento:* | Il sistema mostra un messaggio di errore | |
| Evento innescante | Aver selezionato la voce del menu: “Modifica dati” | | |
| Scenario di base | Il funzionamento del programma è quello atteso, si modificano i dati selezionati del giocatore | | |
| Scenario alternativo | O l’accesso non è avvenuto con successo, o le modifiche apportate non sono valide. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| B06 | Controllo di una data | | Il programma deve verificare la validità di una data. |
| Pre-Condizioni | Inserimento di un input numerico. | | |
| Post-Condizioni | *Successo:* | La data inserita è valida. | |
| *Fallimento:* | La data non è valida, messaggio di errore. | |
| Evento innescante | Aver inserito una data. | | |
| Scenario di base | Il funzionamento del programma è quello atteso, la data è valida. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, l’input inserito non è valido come data. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| B07 | Stampa su file la lista di clienti | | Il programma deve stampare sul file “clienti.csv” la lista dei clienti. |
| Pre-Condizioni | Avere dei dati sulla lista di clienti. | | |
| Post-Condizioni | *Successo:* | Il sistema stamperà su file la lista. | |
| *Fallimento:* | Il sistema non stamperà su file | |
| Evento innescante | Aver effettuato delle modifiche che richiedono un aggiornamento del file | | |
| Scenario di base | Il funzionamento del programma è quello atteso, viene aggiornato il file con la lista di tutti i clienti. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, la lista non viene stampata- | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| C08 | Acquisizione di una prenotazione | | Il programma deve acquisire una nuova prenotazione. |
| Pre-Condizioni | Aver effettuato l’accesso con successo. | | |
| Post-Condizioni | *Successo:* | Il sistema inserisce una nuova prenotazione nel database. | |
| *Fallimento:* | Il sistema non inserisce la nuova prenotazione. | |
| Evento innescante | Aver selezionato la voce del menu: “Prenota ombrellone” | | |
| Scenario di base | Il funzionamento del programma è quello atteso, la nuova registrazione viene salvata nel database delle prenotazioni. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, gli input vengono ripetuti finché non si creano le condizioni accettabili per un corretto inserimento | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| C09 | Ricerca una data | | Il programma deve ricercare una data ricevuta per input. |
| Pre-Condizioni | Aver acquisito un input formato data | | |
| Post-Condizioni | *Successo:* | Il sistema procede alla ricerca ed al confronto delle date presenti nel database prenotazionioni. | |
| *Fallimento:* | Il sistema mostra un messaggio di errore | |
| Evento innescante | Aver inserito una data di cui si vuole conoscere la disponibilità | | |
| Scenario di base | Il funzionamento del programma è quello atteso, il programma confronta la data con tutte le date nel database. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, l’input data non è valido. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| C10 | Ricerca numero di fila ed ombrellone | | Il programma deve ricercare una numero di fila e di ombrellone ricevuto per input. |
| Pre-Condizioni | Aver acquisito un numero di fila ed un numero di ombrellone | | |
| Post-Condizioni | *Successo:* | Il sistema procede alla ricerca ed al confronto delle file e degli ombrelloni presenti nel database | |
| *Fallimento:* | Il sistema mostra un messaggio di errore | |
| Evento innescante | Aver inserito una fila ed un ombrellone di cui si vuole conoscere l’esistenza | | |
| Scenario di base | Il funzionamento del programma è quello atteso, il programma confronta fila ed ombrellone con tutti i dati realativi nel database. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, l’input non è valido. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| C11 | Verifica la disponibilità di una ombrellone | | Il programma deve verificare la disponibilità di un ombrellone. |
| Pre-Condizioni | Aver acquisito un numero di ombrellone | | |
| Post-Condizioni | *Successo:* | Il sistema procede alla verifica della disponibilità di quell’ombrellone. | |
| *Fallimento:* | Il sistema mostra un messaggio di errore | |
| Evento innescante | Aver inserito un ombrellone di cui si vuole conoscere la disponibilità | | |
| Scenario di base | Il funzionamento del programma è quello atteso, il programma confronta fila ed ombrellone con tutti i dati realativi nel database. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, l’input non è valido. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| C12 | Acquisizione tipo di prenotazione | | Il programma deve acquisire un tipo di prenotazione. |
| Pre-Condizioni | Aver inserito un input valido | | |
| Post-Condizioni | *Successo:* | Il sistema procede ad acquisire il tipo di prenotazione | |
| *Fallimento:* | Il sistema mostra un messaggio di errore | |
| Evento innescante | Aver scelto di acquisire una nuova prenotazione | | |
| Scenario di base | Il funzionamento del programma è quello atteso, il programma chiede all’utente di inserire un tipo di prenotazione, acquisendolo. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, l’input non è valido. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| C13 | Acquisizione numero di lettini | | Il programma deve acquisire un numero di lettini. |
| Pre-Condizioni | Aver inserito un input valido | | |
| Post-Condizioni | *Successo:* | Il sistema acquisisce il numero di lettini | |
| *Fallimento:* | Il sistema mostra un errore | |
| Evento innescante | Aver scelto di aggiungere dei lettini | | |
| Scenario di base | Il funzionamento del programma è quello atteso, il sistema chiede quanti lettini aggiungere , salvando i dati. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, l’input non è valido. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| C14 | Acquisizione numero di sedie | | Il programma deve acquisire un numero di sedie. |
| Pre-Condizioni | Aver inserito un input valido | | |
| Post-Condizioni | *Successo:* | Il sistema acquisisce il numero di sedie | |
| *Fallimento:* | Il sistema mostra un errore | |
| Evento innescante | Aver scelto di aggiungere delle sedie | | |
| Scenario di base | Il funzionamento del programma è quello atteso, il sistema chiede quante sedieaggiungere , salvando i dati. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, l’input non è valido. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| C15 | Calcolo del costo totale | | Il programma deve calcolare il costo totale di una prenotazione. |
| Pre-Condizioni | Aver acquisito nella lista dei dati validi | | |
| Post-Condizioni | *Successo:* | Il sistema calcola ed aggiorna il totale della prenotazione | |
| *Fallimento:* | Il sistema non calcola il totale. | |
| Evento innescante | Aver scelto di effettuare delle modifiche ad una prenotazione | | |
| Scenario di base | Il funzionamento del programma è quello atteso, il sistema calcola il nuovo totale. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, l’input non è valido. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| C16 | Stampa a video del tipo di prenotazione | | Il programma deve stampare a video il tipo di prenotazione. |
| Pre-Condizioni | Valori dati validi | | |
| Post-Condizioni | *Successo:* | Il sistema stamperò a video il tipo di prenotazione | |
| *Fallimento:* | Il sistema non mostra il tipo di prenotazione | |
| Evento innescante | Aver scelto di prenotare un ombrellone | | |
| Scenario di base | Il funzionamento del programma è quello atteso, il sistema stampa a video una prenotazione | | |
| Scenario alternativo | La funzione non stampa nulla a video. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| C17 | Stampa su file una nuova prenotazione | | Il programma deve stampare su file una nuova prenotazione. |
| Pre-Condizioni | Avere dei dati di una nuova prenotazione da stampare | | |
| Post-Condizioni | *Successo* | Il sistema stamperà la nuova prenotazione nel csv. | |
| *Fallimento:* | Il sistema non stamperà nulla | |
| Evento innescante | Aver scelto di effettuare una nuova prenotazione | | |
| Scenario di base | Il funzionamento del programma è quello atteso, viene aggiornata la lista con un nuovo campo prenotazione | | |
| Scenario alternativo | Viene mostrato un errore e il file rimane vuoto | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| D18 | Verifica la disponibilità di una fila di ombrelloni | | Il programma deve verificare la disponibilità di una fila di ombrelloni. |
| Pre-Condizioni | Input valido | | |
| Post-Condizioni | *Successo:* | Il sistema verifica della disponibilità di una data fila | |
| *Fallimento:* | Il sistema mostra un messaggio di errore | |
| Evento innescante | Aver selezionato la voce del menu: “Mostra ombrelloni " liberi in un dato giorno scelta una fila” | | |
| Scenario di base | Il funzionamento del programma è quello atteso, viene verificata la disponibilità di una fila. | | |
| Scenario alternativo | Il numero di fila inserito non è valido , il sistema non effettua nessuna verifica. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| D19 | Ricerca numero di fila fra le prenotazioni | | Il programma deve ricercare una numero di fila fra le prenotazioni. |
| Pre-Condizioni | Input fila valido, | | |
| Post-Condizioni | *Successo:* | Il sistema ricerca e trova un numero di fila nel database | |
| *Fallimento:* | Il sistema non trova nessuna fila nel database | |
| Evento innescante | Aver inserito un input ,dopo aver selezionato la voce del menu: “Mostra ombrelloni " liberi in un dato giorno scelta una fila” | | |
| Scenario di base | Il funzionamento del programma è quello atteso, viene confrontato ogni numero di fila con quelli nel database. | | |
| Scenario alternativo | Il programma non trova nessun confronto benchè l’input sia corretto. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| D20 | Elabora la disponibilità del giorno | | Il programma deve calcolare la disponibilità di un giorno. |
| Pre-Condizioni | Aver trovata una fila data in input nel database. | | |
| Post-Condizioni | *Successo:* | Il sistema elabora la disponibilità della giornata in base ai valori nel database. | |
| *Fallimento:* | Il sistema mostra un messaggi di errore | |
| Evento innescante | Voler conoscere la disponibilità per un inserimento/modifica nel database. | | |
| Scenario di base | Il funzionamento del programma è quello atteso, viene elaborata ogni disponibilità per una fila. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso,  il programma elabora in modo errato le disponibilità. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| D21 | Stampa a video la fila di ombrelloni disponibili | | Il programma deve stampare a schermo la disponibilità di una fila di ombrelloni. |
| Pre-Condizioni | Dati validi da stampare | | |
| Post-Condizioni | *Successo:* | Il sistema stamperà a video la fila con le disponibilità per ogni ombrellone | |
| *Fallimento:* | Il sistema non stamperà | |
| Evento innescante | Aver eleborato la disponibilità di una fila da stampare | | |
| Scenario di base | Il funzionamento del programma è quello atteso, viene stampata la disponibilità della fila. | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, non viene stampato nulla. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| E22 | Stampa a video le file di ombrelloni, in ordine crescente/decrescente, disponibili | | Il programma deve stampare a schermo la disponibilità di tutti gli ombrelloni. |
| Pre-Condizioni | Scelta del tipo di stampa crescente/decrescente,  aver inserito una data valida  aver verificato la disponibilità di tutti gli ombrelloni | | |
| Post-Condizioni | *Successo:* | Il sistema stamperà tutte le file di ombrelloni con la loro disponibilità | |
| *Fallimento:* | Il sistema non stamperà nulla | |
| Evento innescante | Aver scelto la voce del menu: “Mostra ombrelloni liberi in un dato " "giorno in ordine crescente/decrescente” | | |
| Scenario di base | Il funzionamento del programma è quello atteso,vengono stampati tutte le file di ombrelloni con la loro disponibilità, in ordine crescente/decrescente | | |
| Scenario alternativo | / | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| E23 | Verifica la disponibilità di tutti gli ombrelloni | | Il programma deve verificare la disponibilità di tutti gli ombrelloni. |
| Pre-Condizioni | Avere dati validi | | |
| Post-Condizioni | *Successo:* | Il sistema verificherà la disponibilità di tutti gli ombrelloni | |
| *Fallimento:* | / | |
| Evento innescante | Aver scelto la voce del menu: “Mostra ombrelloni liberi in un dato " "giorno in ordine crescente/decrescente” | | |
| Scenario di base | Il funzionamento del programma è quello atteso, vengono elaborate tutte le disponibilità. | | |
| Scenario alternativo | / | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| F24 | Modifica dati di una prenotazione | | Il programma deve modificare i dati di una prenotazione. |
| Pre-Condizioni | Numero della prenotazione da modificare  Avere dati validi | | |
| Post-Condizioni | *Successo:* | Il sistema modifica i dati della prenotazione | |
| *Fallimento:* | Il sistema mostra un messaggio di errore | |
| Evento innescante | Aver scelto la voce del menu: “Modifica prenotazione” | | |
| Scenario di base | Il funzionamento del programma è quello atteso, il programma modifica la prenotazione. | | |
| Scenario alternativo | / | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| F25 | Stampa a video una prenotazione | | Il programma deve stampare una prenotazione a video. |
| Pre-Condizioni | Avere dei dati di una prenotazione da stampare a video | | |
| Post-Condizioni | *Successo* | Il sistema stamperà la nuova prenotazione a video | |
| *Fallimento:* | Il sistema non stamperà nulla | |
| Evento innescante | Aver scelto di modificare una prenotazione | | |
| Scenario di base | Il funzionamento del programma è quello atteso, viene mostrata la prenotazione | | |
| Scenario alternativo | Viene mostrato un errore | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| F26 | Modifica tipo di prenotazione | | Il programma deve modificare il tipo di prenotazione. |
| Pre-Condizioni | Aver scelto di modificare il tipo.Input valido, verifica la disponibilità sull’input | | |
| Post-Condizioni | *Successo:* | Il sistema modificherà il tipo | |
| *Fallimento:* | Il sistema non effettuerà modifiche | |
| Evento innescante | Aver scelto la voce del menu: “Modifica prenotazione→ Modifica tipo” | | |
| Scenario di base | Il funzionamento del programma è quello atteso, viene modificato il tipo | | |
| Scenario alternativo | Il sistema segnalera l’impossibilità di effetuare modifiche | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| F27 | Modifica il numero di ombrellone | | Il programma deve modificare il numero di prenotazione. |
| Pre-Condizioni | Aver scelto di modificare il numero di ombrellone  Verifica delle disponibilità | | |
| Post-Condizioni | *Successo:* | Il sistema modificherà il numero di ombrellone | |
| *Fallimento:* | Il sistema non effettuerà modifiche | |
| Evento innescante | Aver scelto la voce del menu: “Modifica prenotazione→ Modifica numero ombrellone” | | |
| Scenario di base | Il funzionamento del programma è quello atteso, viene modificato il numero di ombrellone | | |
| Scenario alternativo | Il sistema segnalera l’impossibilità di effetuare modifiche | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| F28 | Aggiorna file prenotazioni | | Il programma deve aggiornare il file “prenotazioni.csv”. |
| Pre-Condizioni | Dati validi nel database e da aggiornare | | |
| Post-Condizioni | *Successo:* | Il sistema riscrive il file | |
| *Fallimento:* | Il sistema non riscrive il file | |
| Evento innescante | Aver effettuato modifiche sulle prenotazioni | | |
| Scenario di base | Il funzionamento del programma è quello atteso, il file prenotazioni è aggiornato. | | |
| Scenario alternativo | Aggiornamento non avvenuto con successo | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| G29 | Rimozione di una prenotazione | | Il programma deve rimuovere una prenotazione scelta. |
| Pre-Condizioni | Aver selezionato una prenotazione valida da rimuovere | | |
| Post-Condizioni | *Successo:* | Il sistema rimuove la prenotazione scelta dalla lista | |
| *Fallimento:* | Il sistema non rimuove nulla | |
| Evento innescante | Aver scelto la voce del menu: “Elimina prenotazione” | | |
| Scenario di base | Il funzionamento del programma è quello atteso, la prenotazione scelta viene eliminata dalla lista | | |
| Scenario alternativo | / | | |

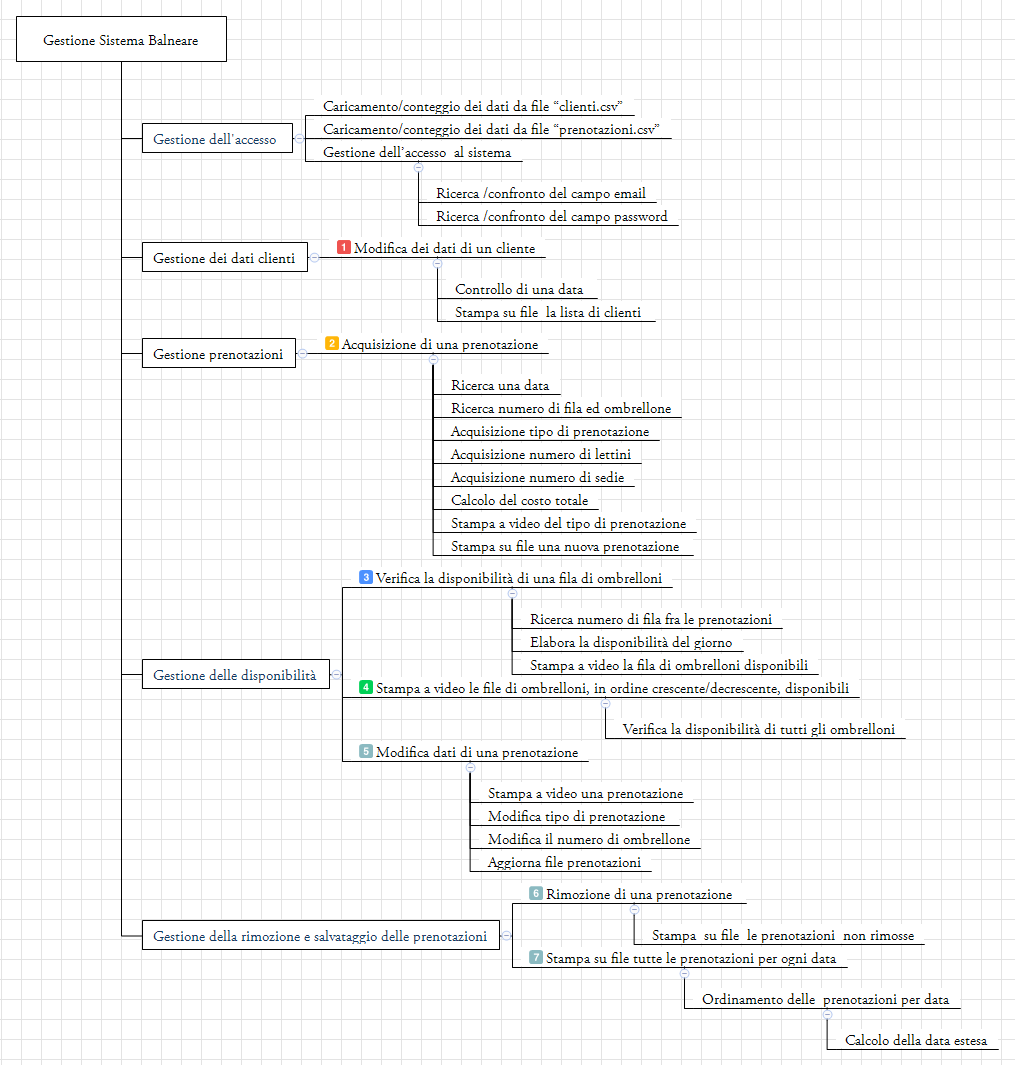
|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| G30 | Stampa su file le prenotazioni non rimosse | | Il programma deve stampare le prenotazioni non rimosse dalla lista. |
| Pre-Condizioni | Aver rimosso con successo una prenotazione dalla lista | | |
| Post-Condizioni | *Successo:* | Il sistema stampa su file la neo-lista | |
| *Fallimento:* | Il sistema non stampa nulla su file | |
| Evento innescante | Aver rimosso dalla lista una prenotazione | | |
| Scenario di base | Il funzionamento del programma è quello atteso, la lista con la rimozione viene stampata su file. | | |
| Scenario alternativo | / | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| H31 | Stampa su file tutte le prenotazioni per ogni data | | Il programma deve stampare su file tutte le prenotazioni divise per ogni data. |
| Pre-Condizioni | Avere dei dati validi nel database | | |
| Post-Condizioni | *Successo:* | Il sistema stamperà un csv per ogni data all’interno delle quali ci saranno le prenotazioni di quel giorno | |
| *Fallimento:* | Il sistema non stamperà nessun file | |
| Evento innescante | Aver scelto la voce del menu: “Salva le prenotazioni di un giorno in un file” | | |
| Scenario di base | Il sistema stamperà un csv per ogni data all’interno delle quali ci saranno le prenotazioni di quel giorno | | |
| Scenario alternativo | / | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| H32 | Ordinamento delle prenotazioni per data | | Il programma deve ordinare tutte le le prenotazioni per data. |
| Pre-Condizioni | Avere dei dati validi nel database | | |
| Post-Condizioni | *Successo:* | Il sistema ordina la lista delle prenotazioni per data | |
| *Fallimento:* | Il sistema non ordina la lista delle prenotazioni | |
| Evento innescante | Aver scelto la voce del menu: “Salva le prenotazioni di un giorno in un file” | | |
| Scenario di base | Il funzionamento del programma è quello atteso, la lista delle prenotazioni è ordinata per data | | |
| Scenario alternativo | Il funzionamento del programma non è quello atteso, la lista delle prenotazioni non è ordinata per data. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Codice | Nome | | Descrizione |
| H33 | Calcolo della data estesa | | l programma deve calcolare il valore di una data per esteso. |
| Pre-Condizioni | Avere dati validi nel database | | |
| Post-Condizioni | *Successo:* | Il sistema calcolerà la data estesa | |
| *Fallimento:* | / | |
| Evento innescante | Aver scelto la voce del menu: “Salva le prenotazioni di un giorno in un file” | | |
| Scenario di base | Il programma calcolerà la data estesa | | |
| Scenario alternativo | / | | |

# 2. Progettazione



## 2.1 Progettazione dei tipi di dato, delle strutture dati

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipologia** | **Descrizione** | **Tipi/Campi/Valori** |
| **data\_t** | struct | Tipo di dato definito per descrivere il campo data | **Giorno :**  int giorno  **Mese :**  int mese  **Anno :**  **int anno** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipologia** | **Descrizione** | **Tipi/Campi/Valori** |
| **ombrellone\_t** | struct | Tipo di dato definito per descrivere il campo ombrellone | **Ombrellone :**  int ombrellone  **Fila ombrellone:**  int fila\_o  **Conta doppia prenotazione :**  **int doppia\_prenotazione**  **Disponibilità :**  **int check** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipologia** | **Descrizione** | **Tipi/Campi/Valori** |
| **cliente\_t** | struct | Tipo di dato definito per descrivere gli attributi dei clienti | **E-mail:**  char email[40]  **Password:**  char password[20]  **Codice fiscale:**  char cod\_fiscale[20]  **Nome:**  char nome [20]  **Cognome:**  char cognome[20]  **Data :**  data\_t nascita  **Sconto :**  double sconto |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipologia** | **Descrizione** | **Tipi/Campi/Valori** |
| **prenotazione\_t** | struct | Tipo di dato definito per descrivere gli attributi delle prenotazioni | **Codice:**  int cod  **E-mail:**  char email[40]  **Data :**  data\_t data  **Tipo :**  int tipo  **Ombrellone:**  ombrellone\_t num\_ombrellone  **Lettini :**  int num\_lettini  **Sedie :**  int num\_sedie  **Costo totale:**  double costo |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipologia** | **Descrizione** | **Tipi/Campi/Valori** |
| **pren\_ombr\_disponibile\_t** | struct | Tipo di dato definito per descrivere tutti gli ombrelloni/file | ombrellone\_t ombrelloni[35]; |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipologia** | **Descrizione** | **Tipi/Campi/Valori** |
| **max\_ombrelloni** | Costante | Tipo di dato definito per indicare il numero massimo di ombrelloni per fila | 35 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipologia** | **Descrizione** | **Tipi/Campi/Valori** |
| **max\_file** | Costante | Tipo di dato definito per indicare il numero massimo di file | 10 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipologia** | **Descrizione** | **Tipi/Campi/Valori** |
| **anno\_corrente** | Costante | Tipo di dato definito per indicare l’anno corrente | 2018 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipologia** | **Descrizione** | **Formattazione** |
| **clienti.csv** | File | File con tutti i dati dei clienti | email,password,codice\_fiscale,nome,cognome,  giorno,mese,anno,sconto |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipologia** | **Descrizione** | **Formattazione** |
| **prenotazioni.csv** | File | File con tutti i dati sulle prenotazioni | codice,email,giorno,mese,anno,tipo,ombrellone,fila,  lettini,sedie,costo |

## 2.2 Progettazione delle librerie / funzioni

Sono state progettate e realizzate 5 librerie:

* f\_all\_shared.h
* f\_disp.h
* f\_edit\_client.h
* f\_new\_pren.h
* f\_out\_remove\_pren.h
* main.h

I metodi nella libreria **f\_all\_shared.h** sono:

**double** **login** ( cliente\_t [], **int** );

**int** **leggi\_conta\_csv\_clienti** (cliente\_t []);

**int** **leggi\_conta\_csv\_prenotazioni** (prenotazione\_t [] );

**int** **verifica\_email**(**char** [],cliente\_t [], **int** , **int** \*);

**int** **verifica\_password** (**char** [],cliente\_t [], **int** );

**int** **controllo\_data** (**int** , data\_t );

**void** **stampa\_prenotazione**(prenotazione\_t [], **int** );

**double** **calcola\_totale** (**double** , ombrellone\_t ,**int** , **int** , **int** );

**void** **stampa\_tipo**(**int** );

**int** **confronta\_prenotazione\_data\_input** (prenotazione\_t [], **int** , data\_t );

**int** **controlla\_tipo\_giornata** ( **int** , **int** );

**void** **stampa\_ombrelloni\_liberi** (pren\_ombr\_disponibile\_t );

**void** **verifica\_disp\_tutti\_ombrelloni** (prenotazione\_t [], data\_t , **int** ,**int** ,**int** ,ombrellone\_t [] );

**void** **stampa\_a\_video\_prenotazione**(prenotazione\_t);

I metodi nella libreria **f\_disp.h** sono:

**void** **ombrelloni\_disponibili\_fila** (prenotazione\_t [], **int** );

**int** **confronta\_prenotazione\_fila** (prenotazione\_t [], ombrellone\_t , **int** );

**void** **ombrelloni\_disponibili\_fila\_cresc\_decr** (prenotazione\_t [], **int** );

**void** **modifica\_prenotazione\_tipo\_ombrellone**(prenotazione\_t [], **int** , **int** , **int** , cliente\_t [] );

**void** **modifica\_tipo\_prenotazione**(prenotazione\_t [], **int** , **int** );

**void** **modifica\_num\_ombrellone\_prenotazione** (prenotazione\_t [] , **int**, **int**);

I metodi nella libreria **f\_edit\_client.h** sono:

**void** **modifica\_dati** (cliente\_t [], **int** , **int** );

**void** **stampa\_clienti** ( cliente\_t [], **int** );

I metodi nella libreria **f\_new\_pren.h** sono:

**void** **prenota\_ombrellone**(prenotazione\_t [] , **int** ,cliente\_t [] , **int** );

**int** **confronta\_prenotazione\_num\_ombrellone\_fila** (prenotazione\_t [],ombrellone\_t );

**void** **stampa\_nuova\_prenotazione** (cliente\_t [], **int** , data\_t, ombrellone\_t, prenotazione\_t [], **int** ,**int** , **int** , **int** , **double**) ;

**int** **tipo\_mpg** ();

**int** **lettini** ();

**int** **sedie** ();

I metodi nella libreria **f\_out\_remove\_pren.h** sono:

**void** **rimuovi\_prenotazione**(prenotazione\_t [], **int** , cliente\_t );

**void** **stampa\_rimuovi\_prenotazione** (prenotazione\_t [], **int** , **int**);

**void** **salva\_prenotazioni\_per\_data**();

**void** **ordinamento\_prenotazioni\_data** (prenotazione\_t [], **int** );

**int** **data\_estesa** (**int** , **int** , **int** );

Le librerie sono state create utilizzando un criterio di separazione delle competenze in: gestione acceso e funzioni condivise nel file **f\_all\_shared.h**, gestione delle disponibilità in **f\_disp.h**, gestione dei dati cliente in **f\_edit\_client.h**, gestione delle nuove prenotazioni **f\_new\_pren.h**, gestione della rimozione e del salvataggio delle prenotazioni **f\_out\_remove\_pren.h**. A questo si aggiunge la libreria **main.h** contenente le strutture e le variabili globali. **Per ulteriori informazioni riguardanti la libreria si invita a controllare la documentazione** [Doxygen](file:///html/index.html).

## 2.3 Dipendenza tra funzioni

Le funzioni che aggiornano gli array delle rispettive struct (clienti e prenotazioni) richiamano la funzione di conta linee e di carica della rispettiva struct, restituendo le righe.

**int** **leggi\_conta\_csv\_clienti** (cliente\_t []);

**int** **leggi\_conta\_csv\_prenotazioni** (prenotazione\_t [] );

La funzione di gestione degli accessi deve accertarsi che l’email e la password inserite siano presenti nel database precedetemene acquisito da file.csv :

**double** **login** ( cliente\_t [], **int** );

Verranno quindi successivamente richiamate per effettuare i controlli :

**int** **verifica\_email**(**char** [],cliente\_t [], **int** , **int** \*);

**int** **verifica\_password** (**char** [],cliente\_t [], **int** );

Un altra funzione spesso richiamata è quella sulla validazione della data:

**int** **controllo\_data** (**int** , data\_t );

Come anche le funzioni di stampa (sia a video che su file) :

**void** **stampa\_prenotazione**(prenotazione\_t [], **int** );

**void** **stampa\_tipo**(**int** );

**void** **stampa\_ombrelloni\_liberi** (pren\_ombr\_disponibile\_t );

**void** **stampa\_a\_video\_prenotazione**(prenotazione\_t);

La funzione di calcolo del totale sarà richiamata da più moduli. Questo per aggiornare i costi dopo modifiche o per il calcolo del costo di una nuova prenotazione:

**double** **calcola\_totale** (**double** , ombrellone\_t ,**int** , **int** , **int** );

Per quanto riguarda la gestione delle disponibilità, avremo la funzione di “calcolo” delle disponibilità (sia sul tipo che sul numero di ombrellone) di tutti gli ombrelloni e la stampa di questi:

**int** **controlla\_tipo\_giornata** ( **int** , **int** );

**void** **verifica\_disp\_tutti\_ombrelloni** (prenotazione\_t [], data\_t , **int** ,**int** ,**int** ,ombrellone\_t [] );

I

Infine, un importante dipendenza avviene nell’ultima funzione del programma riguardante la stampa delle prenotazioni per ogni data.

Infatti la stampa su file richiederà dei confronti, che grazie un ordinamento potremmo effettuare agevolmente. Ma anche lo stesso ordinamento richiede che il formato data venga convertito in un numero scritto per esteso.

Perciò avremo l’interdipendenza di queste tre funzioni:

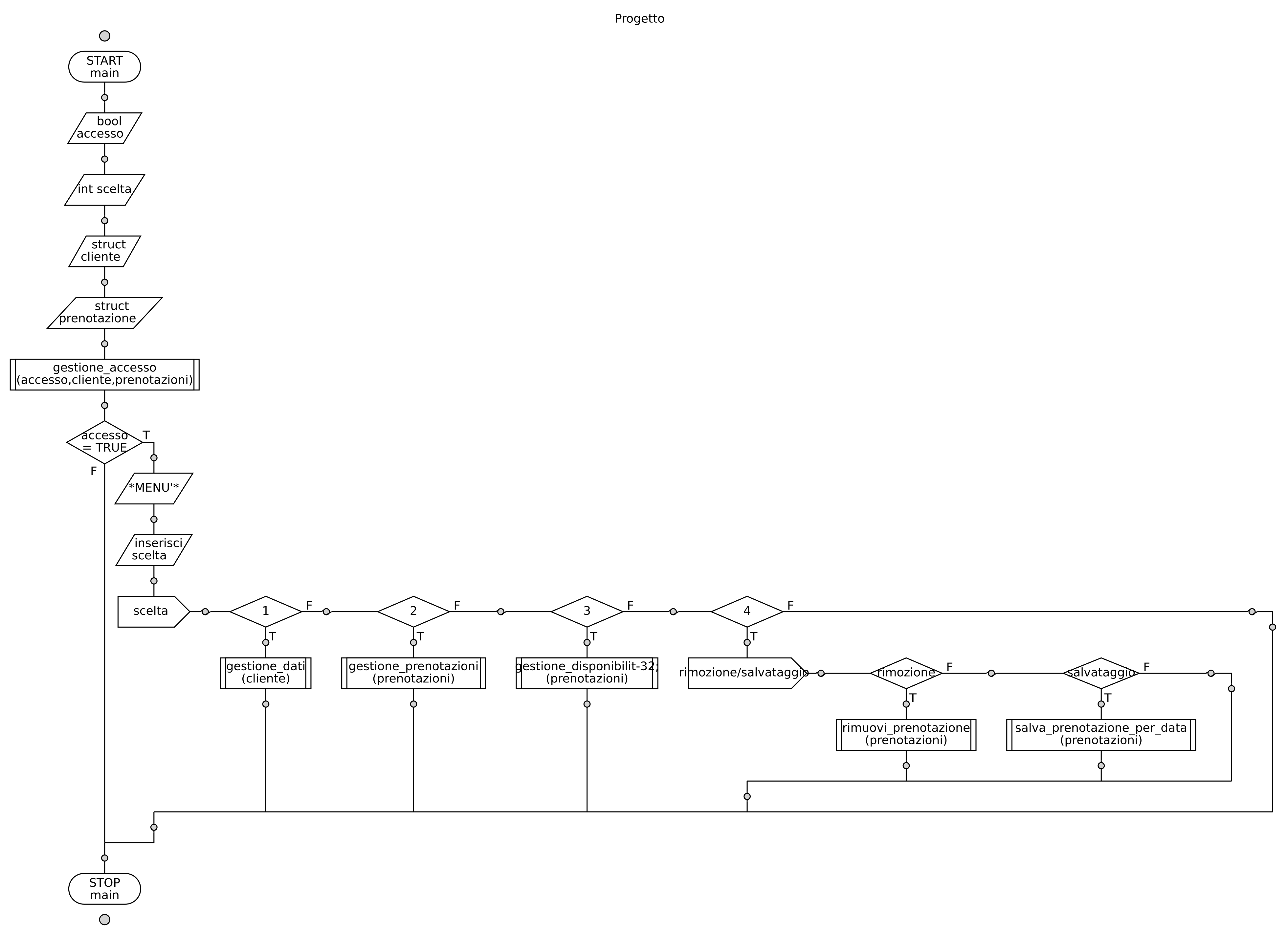
**void** **salva\_prenotazioni\_per\_data**();

**void** **ordinamento\_prenotazioni\_data** (prenotazione\_t [], **int** );

**int** **data\_estesa** (**int** , **int** , **int** );

2.4 Flow-Chart

Main :



# Gestione degli accessi :

# 

# Gestione dei dati :

# 

# Gestione delle prenotazioni :

# Gestione delle disponibilità :

# 

# disponibilità\_fila :

# 

# modifica\_prenotazione :

# 

# Gestione della rimozione e del salvataggio :

# rimuovi\_prenotazione :

# 

# salva\_per\_data :

# 

# 3. Codifica

La documentazione è stata generata attraverso Doxygen ed arricchita con commetti manuali.

Si prega di fare riferimento ai file presenti nella directory del progetto.

# 4. Testing

## 4.1 Definizione del Piano di Test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Codice requisito** | **Codice Test** | **Nome** | **Descrizione Test** | **Eventuale Input** | **Risultato Atteso** | **Risulto Ottenuto** |
| A00 | 1.0 | Caricamento/conteggio dei dati da file “clienti.csv” | File non esistente | - | Messaggio di errore | Messaggio di errore |
| A00 | 1.1 | Caricamento/conteggio dei dati da file “clienti.csv” | File esistente | - | Caricamento dei dati | Caricamento dei dati |
| A00 | 1.2 | Caricamento dei dati da file | File esistente ma vuoto | - | Caricamento dati | Caricamento dati |
| A01 | 1.0 | Caricamento/conteggio dei dati da file “prenotazioni.csv” | File non esistente | - | Messaggio di errore | Messaggio di errore |
| A01 | 1.1 | Caricamento/conteggio dei dati da file “prenotazioni.csv” | File esistente | - | Caricamento dei dati | Caricamento dei dati |
| A01 | 1.2 | Caricamento/conteggio dei dati da file “prenotazioni.csv” | File esistente ma vuoto | - | Caricamento dati | Caricamento dati |
| A02 | 1.0 | Gestione dell’accesso al sistemao giocatore | Nickname non presente | Test | Messaggio di errore | Messaggio di errore |
| A02 | 1.1 | Gestione dell’accesso al sistemao giocatore | Nickname presente | matteoluceri@msn.com | Nickname accettato | Nickname accettato |
| A02 | 1.2 | Gestione dell’accesso al sistemao giocatore | Pasword non presente | Test | Messaggio di errore | Messaggio di errore |
| A02 | 1.3 | Gestione dell’accesso al sistemao giocatore | Password presente | meme420 | Password accettata | Password accettata |
| A03 | 1.0 | Ricerca /confronto del campo email | Confronto avvenuto con successo | - | E-mail valida | E-mail valida |
| A03 | 1.1 | Ricerca /confronto del campo email | Confronto non avvenuto con successo | - | E-mail non valida | E-mail non valida |
| A04 | 1.0 | Ricerca /confronto del campo password | Confronto non avvenuto con successo | - | Password valida | Password valida |
| A04 | 1.1 | Ricerca /confronto del campo password | Confronto non avvenuto con successo | - | Password non valida | Password non valida |
| B05 | 1.0 | Modifica dati di un cliente | Modifica dei campi cliente | selezione campo → input campo | Modifica avvenuta con successo | Modifica avvenuta con successo |
| B05 | 1.1.0 | Modifica dati di un cliente | Modifica dei campi cliente | selezione campo → input data | Se campo data corretto: Modifiche avvenute con succeso | Se campo data corretto: Modifiche avvenute con succeso |
| B05 | 1.1.1 | Modifica dati di un cliente | Modifica dei campi cliente | selezione campo → input data | Se campo data non corretto: Errore | Se campo data non corretto: Errore |
| B06 | 1.0 | Controllo di una data | Controllo della data | 32/13/2017 | Messaggio di errore | Messaggio di errore |
| B06 | 1.1 | Controllo di una data | Controllo della data | 31/12/2018 | Controllo avvenuto con successo | Controllo avvenuto con successo |
| B07 | 1.0 | Stampa su file la lista di clienti | Stampa su file csv | - | Messaggio di operazione effettuata | Messaggio di operazione effettuata |
| C08 | 1.0 | Acquisizione di una prenotazione | Inserimento di una data valida e numero ombrellone | 28/10/2022  -  9,35 | Messaggio ombrellone disponibile | Messaggio ombrellone disponibile |
| C08 | 1.1 | Acquisizione di una prenotazione | Inserimento di una data non valida e numero ombrellone | 25/07/2018  -  10,35 | Messaggio ombrellone non disponibile | Messaggio ombrellone non disponibile |
| C09 | 1.0 | Ricerca una data | Inserimento data valida, ma inesistente | 28/10/2022 | Confronto avvenuto con insuccesso | Confronto avvenuto con insuccesso |
| C09 | 1.1 | Ricerca una data | Inserimento data valida,esistente | 25/07/2018 | Confronto avvenuto con successo:  Messaggio di errore | Confronto avvenuto con successo:  Messaggio di errore |
| C09 | 1.2 | Ricerca una data | Inserimento data non valida | 34/25/3564 | Messaggio di errore | Messaggio di errore |
| C10 | 1.0 | Ricerca numero di fila ed ombrellone | Inserimento fila ed ombrellone valida, ma inesistente | 7,12 | Confronto avvenuto con insuccesso | Confronto avvenuto con insuccesso |
| C10 | 1.1 | Ricerca numero di fila ed ombrellone | Inserimento fila ed ombrellone valida,esistente | 10,35 | Confronto avvenuto con successo:  Messaggio di errore | Confronto avvenuto con successo:  Messaggio di errore |
| C10 | 1.2 | Ricerca numero di fila ed ombrellone | Inserimento fila ed ombrellone non valida | 25,85 | Messaggio di errore | Messaggio di errore |
| C11 | 1.0 | Verifica la disponibilità di una ombrellone | Parametri validi | - | Confronto avvenuto con successo | Confronto avvenuto con successo |
| C11 | 1.1 | Verifica la disponibilità di una ombrellone | Parametri non validi | - | Messaggio di errore | Messaggio di errore |
| C12 | 1.0 | Acquisizione tipo di prenotazione | Input valido | 1 (oppure 2,3) | Modifica avvenuta con successo | Modifica avvenuta con successo |
| C12 | 1.1 | Acquisizione tipo di prenotazione | Input non valido | <1, >3 | Messaggio di errore | Messaggio di errore |
| C13 | 1.0 | Acquisizione numero di lettini | Input valido | 2 | Modifica avvenuta con successo | Modifica avvenuta con successo |
| C13 | 1.1 | Acquisizione numero di lettini | Input non valido | aew | 0 | 0 |
| C14 | 1.0 | Acquisizione numero di sedie | Input valido | 3 | Modifica avvenuta con successo | Modifica avvenuta con successo |
| C14 | 1.1 | Acquisizione numero di sedie | Input non valido | sdw | 0 | 0 |
| C15 | 1.0 | Calcolo del costo totale | Parametri validi | - | Modifica avvenuta con successo | Modifica avvenuta con successo |
| C15 | 1.1 | Calcolo del costo totale | Parametri non validi | - | 0 | 0 |
| C16 | 1.0 | Stampa a video del tipo di prenotazione | Parametri validi | - | Stampa a video | Stampa a video |
| C16 | 1.1 | Stampa a video del tipo di prenotazione | Parametri non validi | - | Stampa a video | Stampa a video |
| C17 | 1.0 | Stampa su file una nuova prenotazione | Parametri validi | - | Stampa su csv | Stampa su csv |
| C17 | 1.1 | Stampa su file una nuova prenotazione | Parametri non validi | - | Stampa su csv | Stampa su csv |
| D18 | 1.0 | Verifica la disponibilità di una fila di ombrelloni | Input fila valido | 5 | Chiama procedura controllo | Chiama procedura controllo |
| D18 | 1.1 | Verifica la disponibilità di una fila di ombrelloni | Input fila non valido | 26 | Chiama procedura controllo | Chiama procedura controllo |
| D19 | 1.0 | Ricerca numero di fila fra le prenotazioni | Parametri validi | - | Chiama procedura di verifica disponibilità | Chiama procedura di verifica disponibilità |
| D19 | 1.1 | Ricerca numero di fila fra le prenotazioni | Parametri non validi | - | Messaggio di errore | Messaggio di errore |
| D20 | 1.0 | Elabora la disponibilità del giorno | Avrà solo parametri validi. Il controllo è già avvenuto |  | Stampa nella struttura tutte le disponibilità di una fila e un giorno. Chiama procedura di stampa | Stampa nella struttura tutte le disponibilità di una fila e un giorno. Chiama procedura di stampa |
| D21 | 1.0 | Stampa a video la fila di ombrelloni disponibili | Stampa a video le disponibilità precedetemente elaborate | - | Stampa a video le disponibilità | Stampa a video le disponibilità |
| E22 | 1.0 | Stampa a video le file di ombrelloni, in ordine crescente/decrescente, disponibili | Selezione crescente | 1 | Previo controllo delle disponibilità:Stampa a video le disponibilità di tutti gli ombrelloni in ordine crescente di fila | Previo controllo delle disponibilità:Stampa a video le disponibilità di tutti gli ombrelloni in ordine crescente di fila |
| E22 | 1.1 | Stampa a video le file di ombrelloni, in ordine crescente/decrescente, disponibili | Selezione decrescente | 2 | Previo controllo delle disponibilità:Stampa a video le disponibilità di tutti gli ombrelloni in ordine decrescente di fila | Previo controllo delle disponibilità:Stampa a video le disponibilità di tutti gli ombrelloni in ordine decrescente di fila |
| E23 | 1.0 | Verifica la disponibilità di tutti gli ombrelloni | - | - | Elabora disponibilità di tutti gli ombrelloni | Elabora disponibilità di tutti gli ombrelloni |
| F24 | 1.0 | Modifica dati di una prenotazione | Scelta di una prenotazione tra quelle mostrate | x, dove x è il numero della prenotazione da modificare | Chiama procedura di modifica | Chiama procedura di modifica |
| F24 | 1.1 | Modifica dati di una prenotazione | Scelta di una prenotazione tra quelle mostrate | x, dove x non appartiene al numero delle prenotazioni da modificare | Messaggio di errore | Messaggio di errore |
| F25 | 1.0 | Stampa a video una prenotazione | Procedurà di stampa | - | Stampa a video una prenotazione | Stampa a video una prenotazione |
| F26 | 1.0 | Modifica tipo di prenotazione | Previa verifica della disponibilità avvenuta con successo: Input valido | M,P,G | Modifiche avvenute con successo | Modifiche avvenute con successo |
| F26 | 1.1 | Modifica tipo di prenotazione | Previa verifica della disponibilità avvenuta con insuccesso: Input valido | M,P,G | Messaggio di errore | Messaggio di errore |
| F27 | 1.0 | Modifica il numero di ombrellone | Previa verifica della disponibilità avvenuta con successo: Input valido | <35,>0 | Modifiche avvenute con successo | Modifiche avvenute con successo |
| F27 | 1.1 | Modifica il numero di ombrellone | Previa verifica della disponibilità avvenuta con successo: Input non valido | >35,<0 | Messaggio di errore | Messaggio di errore |
| F27 | 1.2 | Modifica il numero di ombrellone | Previa verifica della disponibilità avvenuta con insuccesso: Input valido | <35,>0 | Messaggio di errore | Messaggio di errore |
| F28 | 1.0 | Aggiorna file prenotazioni | Procedurà di aggiornamento prenotazioni.csv,parametri validi | - | Stampa su csv | Stampa su csv |
| F28 | 1.1 | Aggiorna file prenotazioni | Procedurà di aggiornamento prenotazioni.csv,parametri non validi | - | Stampa su csv | Stampa su csv |
| G29 | 1.0 | Rimozione di una prenotazione | Seleziona numero prenotazione | x, dove x non appartiene al numero delle prenotazionei da poter rimuovere | Chiamata procedura di rimozione | Chiamata procedura di rimozione |
| G30 | 1.0 | Stampa su file le prenotazioni non rimosse | Porcedura | - | Stampa su csv tutte le prenotazioni non selezionate in G29 | Stampa su csv tutte le prenotazioni non selezionate in G29 |
| H31 | 1.0 | Stampa su file tutte le prenotazioni per ogni data | Previo ordinamento | - | Stampa un file csv per tutte le prenotazioni di ogni data | Stampa un file csv per tutte le prenotazioni di ogni data |
| H32 | 1.0 | Ordinamento delle prenotazioni per data | Ordinamento delle prenotazioni.csv | - | Previa chiamata procedura data estesa H33, ordina il file | Previa chiamata procedura data estesa H33, ordina il file |
| H33 | 1.0 | Calcolo della data estesa | Paramentri validi | - | Calcolo della data estesa | Calcolo della data estesa |

## 4.2 Esiti del Piano di Test ed eventuali commenti

Il programma, a seguito della fase di test, evidenzia una forte interdipendenza fra alcune funzioni.

Spesso anche il solo malfunzionamento di un modulo, di un file o di un input (e quindi dei parametri delle funzioni/procedure chiamanti) , porterà ad un mal funzionamento a valanga di una parte delle funzionalità del sistema.   
Per ovviare a ciò l’autore propone due soluzioni:

1. Una maggiore e migliore validazione degli input:

Infatti, benché sia stata progettata un importante sistema di validazione - sull’accesso , sul controllo delle date, e meglio ancora quello sulle disponibilità e del tipo di prenotazione– il programma risente dell’assenza della validazione su altri input. Per esempio si crede necessaria una maggiore validazione sul campo codice fiscale, e più in generale su tutti i campi oggetto di modifiche non opportunamente controllate.

1. Una migliore programmazione modulare:

Una migliore modularizzazione avrebbe potuto ovviare ad alcuni problemi. Maggiori controlli nel passaggio di parametri, magari “all’entrata ed uscita” di un modulo.

Nonostante ciò, un corretto utilizzo del programma adempie esattamente ai compiti per cui è stato ideato.

Nel corso d’opera non sono state implementate funzioni che l’autore aveva in parte ideato. In particolare inserire la possibilità di registrarsi. Infatti ciò avrebbe reso il sistema più utilizzabile ed “utile”. Infatti in assenza di una funzione che garantisca la registrazione, un utente generico dovrebbe manipolare direttamente i file, provocando diversi disagi. Insieme a ciò era stata ideata una validazione sull’email, fondamentale per garantire una corretta registrazione. Le funzionalità extra sono state successivamente accantonate a causa di una implementazione tardiva nel sistema, che ne avrebbe causato una parziale ri-progettazione.

Altre pecche invece associabili allo stile di programmazione sono sicuramente:

* L’uso sporadico (unico caso è nell’ultima funzione) di allocazione dinamica all’interno del programma.
* Largo uso di controlli e domande iterative all’utente.

Nonostante ciò renda l’esperienza dell’utente più “agevole” e controllata, ha reso il codice di difficile lettura e confusionario in alcuni punti, complicando oltretutto l’implementazione di procedure/funzioni associate , nel corso d’opera. Per ovviare a ciò l’autore ha opportunamente commentato tutte le fasi di controllo e le condizioni di uscita.

Concludendo, un ottimizzazione del sistema orientata ad una maggiore validazione degli input, ad un miglior uso della memoria e dei moduli, inseme l’implementazione di funzionalità aggiuntive - quali possono essere la registrazione o la possibilità di effettuare abbonamenti – renderebbero il programma ideale al raggiungimento degli obbiettivi funzionali definiti e richiesti nella traccia.

*Matteo Luceri*