DIAGRAMMA DEI CASI D’USO

PROGETTO DI FINE ANNO 2020/2021 DI MICHELE GRESSANI CENTRO FITNESS

Salvare i dati su un file binario

Esportare in formato CSV tutti gli accessi

Aggiungere l’accesso di un utente

Registrare le uscite dalla struttura aggiungendo data e ora di uscita

Eliminare un accesso in base al nome e alla data di accesso

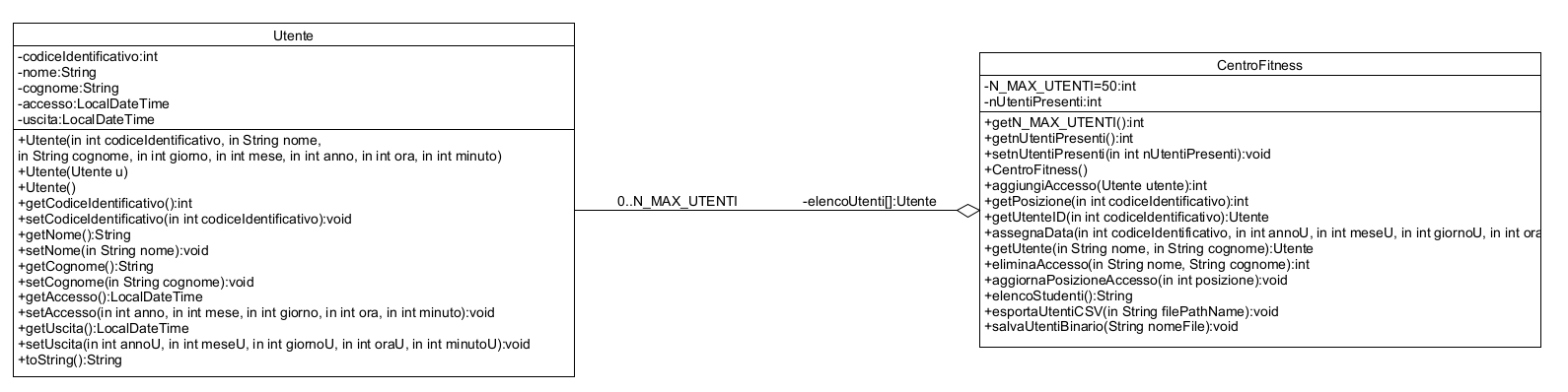
Visualizzare data e ora di entrata e, se c’è, uscita di tutti gli accessi di un cliente in base al nome e cognome

Visualizzare tutti gli accessi effettuati

TABELLA DEI REQUISITI

|  |  |  |  |
| --- | --- | --- | --- |
| N° | Requisito | Tipologia | Importanza |
| 1 | Registrare gli accessi alla struttura | Non funzionale | MUST |
| 2 | Ogni utente deve registrare: codice ID, nome, cognome, data e ora di entrata e uscita | Non funzionale | MUST |
| 3 | Aggiungere un accesso inserendo codice ID, nome, cognome, data e ora di entrata | Funzionale | MUST |
| 4 | Registrare l’uscita dalla struttura registrando data e ora di uscita | Funzionale | MUST |
| 5 | Eliminare un accesso in base al nome e alla data di accesso | Funzionale | MUST |
| 6 | Dato un cognome e un nome visualizzare quando quella persona è stata presente | Funzionale | MUST |
| 7 | Visualizzare tutti gli accessi effettuati | Funzionale | MUST |
| 8 | Esportare tutti gli accessi in formato CSV | Tecnologico | MUST |
| 9 | Salvare tutti i dati su un file binario | Funzionale | MUST |
| 10 | Caricare i dati all’avvio del progetto | Non funzionale | MUST |

DIAGRAMMA DELLE CLASSI



TEST DI INTEGRAZIONE

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Classe CentroFitness | | | | | | | | |
| N | | DESCRIZIONE | | PRECONDIZIONI | AZIONI | | RISULATO  ATTESO | ESITO  NEGATIVO |
| getNUM\_MAX\_UTENTI() | | | | | | | | |
| 1 | |  | | Istanzio un Utente vuoto | c1.getNUM\_MAX\_UTENTI() | | 50 |  |
| getnUtentiPresenti() | | | | | | | | |
| 2 | |  | | Istanzio un utente vuoto | c1.getnUtentiPresenti() | | 0 |  |
| 3 | |  | | Aggiungo u1 | c1.getnUtentiPresenti() | | 1 |  |
| 4 | |  | | Aggiungo u2 | c1.getnUtentiPresenti() | | 2 |  |
| 5 | |  | | Rimuovo u1 | c1.getnUtentiPresenti() | | 1 |  |
| getPosizione(int posizione) | | | | | | | | |
| 6 | | getPosizione  utente presente | | Sono presenti 6  utenti | getPosizione(4) | | Oggetto utente in posizione 4 |  |
| 7 | | getPosizione  utente non presente | | Sono presenti  6 utenti | getPosizione(10) | | Non ci sono utenti  null |  |
| getUtenteID(int codiceIdentificativo) | | | | | | | | |
| 8 | | getUtente  utente presente | | Sono presenti  6 codici | getUtenteID(3) | | Codice identificativo dell’utente |  |
| 9 | | getUtente  utente presente | | Sono presenti  6 codici | getUtenteID(8) | | Non c’è il codice |  |
| aggiungiAccesso(Utente utente) | | | | | | | |  |
| 10 | | Aggiunta di un utente | | Istanzio un utente | CentroFitness a  c1.aggiungiAccesso(u1) | | Return 0; |  |
| 11 | | Aggiunta di un utente  Ma nUtentiPresenti è >= a NUM\_MAX\_UTENTI | | Istanzio un utente a  con  nUtentiPresenti è >= a NUM\_MAX\_UTENTI | CentroFitness a  c1.aggiungiUtente(u84) | | Il numero massimo di utenti è stato raggiunto |  |
| getUtente(String nome, String cognome) | | | | | | | | |
| 12 | | getUtente  utente presente | | Sono presenti  5 utenti | getUtente(nome, cognome) | | Oggetto utente con nome e cognome corrispondenti |  |
| 13 | | getUtente  utente non presente | | Sono presenti  5 utenti | getUtente(nome, cognome) | | Non ci sono utenti  null |  |
| elencoUtenti(String targa) | | | | | | | | |
| 14 | | Visualizzo i dati di tutti gli utenti esistenti | | Istanzio un utente u1  codiceIdentificativo:1  Nome:michele  Cognome:gressani  Giorno: 12  Mese: 05  Anno: 2021  Ora: 15  Minuto: 45 | c1.elencoUtenti() | | Istanzio un utente u1  codiceIdentificativo:1  Nome:michele  Cognome:gressani  Giorno: 12  Mese: 05  Anno: 2021  Ora: 15  Minuto: 45 |  |
| assegnaData(int codiceIdentificativo, int giornoU, int meseU, int annoU, int oraU, int minutoU) | | | | | | | | |
| 15 | Assegno ad un utente esistente  la data di uscita | | Istanzio un utente u1 con codiceIdentificativo 1  Giorno: 12  Mese: 05  Anno: 2021  Ora: 15  Minuto: 45 | | | c1.assegnaData(1, 12, 05, 2021, 15, 45) | Utente con data e ora di uscita:  giornoU: 12  meseU: 05  annoU: 2021  oraU: 15  minutoU: 45 |  |
| 16 | Assegno ad un utente inesistente  la data di uscita | | Istanzio un utente u1 con codiceIdentificativo: 1  Giorno: 12  Mese: 05  Anno: 2021  Ora: 15  Minuto: 45 | | | c1.assegnaData(1, 12, 05, 2021, 15, 45) | Il codiceIdentificativo inserito non è esistente |  |
| eliminaAccesso(String nome, LocalDateTime accesso) | | | | | | | | |
| 17 | Elimina utente presente | | Istanzio 3 utenti con i seguenti nomi e la seguente data di accesso michele 12/05/2021 8:00, andrea 14/06/2021 19:30, giovanni 19/12/2021 20:15 | | | eliminaUtente(michele, 12, 05, 2021, 8, 00) | eliminazione avvenuta correttamente |  |
| 18 | Elimina utente non presente | | Istanzio 3 utenti con i seguenti nomi e la seguente data di accesso michele 12/05/2021 8:00, andrea 14/06/2021 19:30, giovanni 19/12/2021 20:15 | | | eliminaUtente(nicola 13, 12, 2021, 7, 50) | Il nome e accesso non sono esistenti |  |
| aggiornaPosizioneAccesso(int posizione) | | | | | | | | |
| 20 | Sposto la posizione dell’utente 4 in 3 | | Istanzio 4 utenti | | | aggiornaPosizioneRevisione(4) | l’array è stato modificato |  |
| esportaUtentiCSV (String nomeFile) | | | | | | | | |
| 21 | Salvo su file csv il contenuto dell’array di oggetti | | Instanzio un oggetto di classe CentroFitness | | | esportaUtentiCSV (“utentiCentroFitness.txt”) | Salvataggio avvenuto correttamente |  |
| 22 | Salvo su file csv il contenuto dell’array di oggetti, ma il file non è di tipo txt | | Instanzio un oggetto di classe CentroFitness | | | salvaRevisione(“utentiCentroFitness.av”) | impossibile accedere al file, gli utenti non sono stati salvati |  |
| salvaUtentiBinario (String nomeFile) | | | | | | | | |
| 23 | Salvo su file binario il contenuto dell’array di oggetti | | Instanzio un oggetto di classe CentroFitness | | | salvaUtentiBinario (“utentiCentroFitnessBinario.bin”) | Salvataggio avvenuto correttamente |  |
| 24 | Salvo su file binario il contenuto dell’array di oggetti, ma il nome del file non è .bin | | Instanzio un oggetto di classe CentroFitness | | | salvaUtentiBinario (“utentiCentroFitnessBinario.txt”) | impossibile accedere al file binario, gli utenti non sono stati salvate |  |

TEST UNITARIO

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N | DESCRIZIONE | PRECONDIZIONI | AZIONI | RISULTATO ATTESO | ESITO NEGATIVO | osservazioni |
| 1 | registrare un nuovo accesso | Utente vuoto | Menu case 1,  inserimento attributi | Ritorno al menu iniziale |  |  |
| 2 | Registrazione di un nuovo utente con data non numerica |  | Menù case 1,  inserimento attributi giorno: giovanni  mese: 12  anno:2021 | Input non corretto ritorna al menu iniziale |  |  |
| 3 | Visualizza la data di presenza di uno specifico utente | Dopo aver eseguito il test numero 1 | Menù case 3,  inserimento nome=”michele” e cognome=”gressani” | Visualizzazione della data di presenza dell’utente con nome “michele” e cognome “gressani” |  |  |
| 4 | Visualizza la data di presenza di uno specifico utente |  | Menù case 3,  inserimento nome”giovanni” e cognome “andreoli” | Utente non presente, Ritorno al menu iniziale |  |  |
| 5 | Visualizza tutti gli accessi | Dopo aver eseguito il test numero 1 | Menù case 5 | Visualizzazione di tutti gli utenti |  |  |
| 6 | Visualizza tutti gli accessi senza utenti presenti |  | Menù case 5 | Nessun utente presente, Ritorno al menu iniziale |  |  |
| 7 | Registra uscita utente esistente | Dopo aver eseguito il test numero 1 | Menù case 4,  inserimento codiceIdentificativo esistente | Inserimento data uscita,  ritorno al menu iniziale |  |  |
| 8 | Registra uscita utente inesistente |  | Menù case 4,  inserimento codiceIdentificativo inesistente | Utente non presente, ritorno al menu iniziale |  |  |
| 8 | elimina un utente aggiunto | Dopo aver eseguito il test numero 1 | Menù case 2,  inserisco il nome e la data di accesso di un utente esistente | Accesso rimoso correttamente, Ritorno al menu iniziale |  |  |
| 9 | elimina un utente non esistente |  | Menù case 2,  inserisco il nome e la data di accesso di un utente inesistente | Nessun utente presente,  ritorno al menu iniziale |  |  |
| 10 | Esportare in formato CSV tutti gli interventi | Dopo aver effettuato il test numero 1 | Menù case 6 | Utenti esportati correttamente, creazione di un file di testo ”utentiCentroFitness.txt” , Ritorno al menu iniziale |  |  |
| 11 | Salva utenti su file binario | Dopo aver effettuato il test numero 1 | Menù case 7 | salvataggio avvenuto correttamente, creazione di un file binario ”utentiCentroFitnessBinario.bin” , Ritorno al menu iniziale |  |  |
| 12 | Selezione voce del menu negativa |  | Scelta menu -13 | Valore inserito non valido. Premi invio ed effettua nuovamente la scelta, Ritorno al menu iniziale |  |  |
| 13 | Selezione voce del menu non esistente |  | Scelta menu 9 | Valore inserito non valido. Premi invio ed effettua nuovamente la scelta, Ritorno al menu iniziale |  |  |
| 14 | Selezione voce del menu non numerica |  | Scelta menu adsf | Valore inserito non valido. Premi invio ed effettua nuovamente la scelta, Ritorno al menu iniziale |  |  |
| 15 | Esci dal software | Dopo aver effettuato il test numero 14 | Scelta menu 0 | Applicazione chiusa. Arrivederci! :) |  |  |
| 16 | Caricamento corretto dei data | Dopo aver effettuato il test numero 15 | Avvio del software | Lettura avvenuta correttamente, apertura menu principale |  |  |