|  |  |
| --- | --- |
| ***Definizione*** | ***Funzionamento*** |
| Persona/Account | |
| bool Login(string username, string password); | * TRUE: username e password corretti * FALSE: * Username e password sbagliati * Errore nella Query * Connessione non eseguita con il DB * Altra eccezione |
| string GetPrivilegi(string username); | Ritorna I privilege dell ‘Account:   * “direttore” * “cliente” * “impiegato” * String.Empty : errore nell’esecuzione della query o mancata connessione con DB |
| List<Persona> GetListaPersone(string tipoAccount, string idFiliale); | Restituisce una lista di persone che hanno l’account del tipo specificato in tipoAccount (“client”, “impiegato”…) e che appartengono ad una determinata filiale con l’id uguale a idFiliale |
| bool EliminaAccount(string username); | Elimina l’Account selezionato con username.   * TRUE: se l’account viene eliminato correttamente * FALSE: * Username non trovato * Errore nella query * Connessione non eseguita con il DB * Altra eccezione |
| Persona CheckUsername(string username); | Cerca i dati dell’Account e della Persona con l’username uguale a quello dato. Se viene trovato, restituisce l’oggetto Persona contente i dati anagrafici e dell’account. Altrimenti restituisce un’oggetto Persona con tutti gli attributi settati a Default. Questo avviene anche in caso di eccezioni. |
| Persona GetPersona(string codiceFiscale); | Cerca i dati dell’Account e della Persona con il codiceFiscale uguale a quello dato. Se viene trovato, restituisce l’oggetto Persona contente i dati anagrafici e dell’account. Altrimenti restituisce un’oggetto Persona con tutti gli attributi settati a Default. Questo avviene anche in caso di eccezioni. |
| bool AggiungiPersona(Persona persona, string password); | Aggiunge un Account, e aggiunge una Persona se la persona non ha già un altro account.   * TRUE: se l’account viene aggiunto correttamente * FALSE: * Account non aggiunto * Errore nella query * Connessione non eseguita con il DB * Altra eccezione |
| bool ModificaPersona(string identificativo, Persona persona); | INPUT:   * Identificativo: può essere codice fiscale o username dell’Account/Persona da modificare * Persona: contiene le informazioni da aggiornare nel DB. Se un attributo non è da aggiornare viene lasciato al suo valore di default in persona.   Es. persona.CAP = null, non verrà modificato nel DB.  persona.Nome = “Mario”, verrà aggiornato.  persona.Cogome = string.Empty, non verrà aggiornato.  OUTPUT:   * TRUE: profilo modificato correttamente. * FALSE: * Account non modificato * Errore nella query * Connessione non eseguita con il DB * Altra eccezione |
| Conto Corrente | |
| ContoCorrente SelectContoCorrente(int idContoCorrente); | Cerca il conto corrente con l’idContoCorrente uguale a quello dato. Se viene trovato, restituisce l’oggetto ContoCorrente contente i dati del conto. Altrimenti restituisce un’oggetto ContoCorrente con tutti gli attributi settati a Default. Questo avviene anche in caso di eccezioni. |
| bool CheckIBAN(string IBAN); | Controlla se esiste un ContoCorrente con quell’IBAN, valore di ritorno:   * TRUE: * Esiste un ContoCorrente * FALSE: * Non è stato trovato un ContoCorrente con quell’IBAN * Errore nella query * Connessione non eseguita con il DB * Altra eccezione |
| bool CheckIDConto(int idContoCorrente); | Controlla se esiste un ContoCorrente con quell’idContoCorrente, valore di ritorno:   * TRUE: * Esiste un ContoCorrente * FALSE: * Non è stato trovato un ContoCorrente con quell’idContoCorrente * Errore nella query * Connessione non eseguita con il DB * Altra eccezione |
| bool CheckContoCorrente(int idContoCorrente) | **NON SONO STATI IMPLEMENTANTI** |
| ContoCorrente GetContoCorrente(int idContoCorrente) |
| Filiale | |
| Filiale GetFiliale(string username); |  |
| bool ModificaDatiFiliale(string idFiliale, Filiale nuovaFiliale); |  |
| string GetNameFiliale(string idFiliale); |  |
| Movimenti | |
| List<Movimento> GetListaMovimenti(int idContoCorrente); |  |
| bool CheckImporto(decimal importo, string IBANCommittente); |  |
| bool EseguiBonifico(string IBANCommittente, string IBANBeneficiario, decimal Importo); |  |
| bool EseguiPrelievoDenaro(int IBANCommittente, decimal importo); |  |
| bool EseguiDeposito(int IBANCommittente, decimal importo); |  |