FEntityManagerSQL:

realizzato con il pattern singleton:

1)getInstance() per creare l’istanza dell’entity manager

2)getDB() per ritornare l'attributo "$db" che è il PDO che effettua la connessione al dbms

3)closeConnection() per chiudere la connessione distruggendo l’istanza dell’entity manager

Metodi generalizzati per le altre classi:

1)retriveObj($table, $field ,$id) restituisce il risultato della query relativa

SELECT \* FROM table WHERE field id

2)getObjOnAttributes($table, $field1, $id1, $field2, $id2) altra query con 2 campi

SELECT \* FROM table WHERE field1 = id1 AND field2 = id2

3)existInDb($queryResult) necessario alla verifica dell’esistenza di risultati di una query

4)updateObj($table, $field, $fieldValue, $cond, $condValue) per le UPDATE

UPDATE $table SET field =fieldValue WHERE cond =condValue

5)saveObject($foundClass, $obj) per effettuare le INSERT (i metodi sono implementati)

INSERT INTO foundClass::getTable() VALUES foundClass::getValues()

6)saveObjectFromId($foundClass, $obj, $id) aggiungiamo l’id avendo già l’oggetto istanzianto, bisogna Vedere se funziona bene

7) objectListNotRemoved($table, $field, $id) (NON USATO PER ORA)

8) deleteObjInDb($table, $field, $id) Effetuare la query di DELETE seguente

DELETE FROM table WHERE field = id

9) checkCreator($queryResult, $idUser) (NON USATA PER ORA)

10) getSearchedItem($table, $field, $str) (NON USATA PER ORA)

11) objectListOnNull($table, $field) (NON USATA PER ORA)

Metodi comuni per tutte le classi del modello

\_\_construct()

getClass()

getTable()

getValues()

getKey()

bind($stmt, ECLASSE $OGGETTO DELLA CLASSE) per il binding di un oggetto istanziato

risultano efficaci per l’utilizzo nei metodi e la scrittura di query

CLASSI PRIMITIVE (Amministratore,Paziente,Immagine,Tipologia)

Classe FPaziente: (similarità ad FUser)

1) creapaziente($queryResult)->Per fare la load dal db mettendo come argomento il risultato della query con i pazienti

2) getObj($id)->Per ottenere un paziente utilizzando il suo id

3) getpazientefromnome\_cognome($nome,$cognome)->Per ottenere un paziente avendo nome e cognome

4) saveObj ($paziente, $fieldArray = null)->Per salvare sul DB o effettuare un'update a dei valori di un paziente

Classe FAmministratore: (similarità ad FModerator)

1) creaamministratore($queryResult)->Per fare la load dal db mettendo come argomento il risultato della query il record dell'admin

2) getObj($id)->Per ottenere un admin utilizzando il suo id

3) saveObj($admin)-> Per salvare sul DB un amministratore Per ottenere un admin avendo la sua mail

4) getadminbyemail($email)->Per ottenere un admin avendo la sua mail

Classe FImmagine (simile ad FImage)

1)creaimmagine($queryResult)->Per fare la load dal db mettendo come argomento il risultato della query con l'immagine

2) getObj($id)->Per ottenere un'immagine utilizzando il suo id

3) getimmaginefromidmedico($IdMedico)->Per ottenere l'immagine relativa ad un medico

4) getimmaginefromidreferto($IdReferto)-> Per ottenere l'immagine relativa ad un referto

5) saveObj($immagine)->Per salvare sul DB un'immagine

6) eliminaimmagine($IdImmagine) -> eliminare un’immagine dal suo id

Classe FTipologia

1) creatipologia($queryResult)->Per fare la load dal db mettendo come argomento il risultato della query con le tipologie

2) getObj($IdTipologia)->Per ottenere una tipologia utilizzando il suo id

3) saveObj($tipologia)->Per salvare sul DB una tipologia

4) eliminatipologia($nome\_tipologia)->Per eliminare dal DB una tipologia

CLASSI DERIVATE (Medico,Calendario,Fascia Oraria,Appuntamento,Referto,Recensione,Risposta)

Classe FMedico:(similarità ad FUser)

1) creamedico($queryResult)->Per fare la load dal db mettendo come argomento il risultato della query con i medici

2) getObj($id)->Per ottenere un medico utilizzando il suo id

3) getmedicofromtipologia($idTipologia)->Per ottenere una lista di medici di una determinata tipologia

4) saveObj($medico, $fieldArray = null)-> Per salvare sul DB o effettuare un'update a dei valori di un medico

Classe FCalendario:(similarità a POST)

1) creacalendario($queryResult)->Per fare la load dal db mettendo come argomento il risultato della query con il calendario

2) getObj($IdCalendario)->Per ottenere un calendario utilizzando il suo id

3) getcalendariofromidmedico($IdMedico)->Per ottenere un calendario dal suo IdMedico

4) saveObj($calendario , $fieldArray = null)->Per salvare sul DB o effettuare un'update a dei valori di un calendario (le modifiche potrebbero essere tolte)

Classe FFasciaOraria (similarità a POST)

1) creafasciaoraria($queryResult)->Per fare la load dal db mettendo come argomento il risultato della query con il calendario

2) getObj($IdFascia\_oraria)->Per ottenere una fascia oraria utilizzando il suo id

3) saveObj($fasciaoraria)->Per salvare sul DB una fascia oraria

4) eliminafasciaoraria($IdFascia\_oraria)->Per eliminare dal DB una fascia oraria

Classe FAppuntamento (come Recensione)

1) creaappuntamento($queryResult)->Per fare la load dal db mettendo come argomento il risultato della query con le gli appuntamenti

2) getObj($IdAppuntamento)->Per ottenere un appuntamento utilizzando il suo id

3) saveObj($obj, $fieldArray = null)->Per salvare sul DB una un appuntamento o modificarne uno esistente

4) eliminaappuntamento($IdAppuntamento)Per eliminare dal DB una appuntamento

Classe FReferto (simile ad FComment)

1) creareferto($queryResult)->Per fare la load dal db mettendo come argomento il risultato della query con il referto

2) getObj($IdReferto)->Per ottenere un referto utilizzando il suo id

3) saveObj($referto)->Per salvare sul DB una un referto (da aggiungere la parte della modifica per le immagini e gli altri campi?)

4) eliminareferto($IdReferto)->Per eliminare dal DB un referto

Classe FRecensione (similarità ad FPost)

1) crearecensione($queryResult)->Per fare la load dal db mettendo come argomento il risultato della query con le recensioni

2) getrecensionefromid($IdRecensione)->Per ottenere una Recensione utilizzando il suo id

3) salvarecensione($recensione)Per salvare sul DB una recensione

4) eliminarecensione($IdRecensione)->Per eliminare dal DB una recensione

Classe FRisposta (similarità ad FComment)

1) crearisposta($queryResult)->Per fare la load dal db mettendo come argomento il risultato della query con le risposte

2) getrispostafromid($IdRisposta)->Per ottenere una risposta utilizzando il suo id

3) salvarisposta($risposta)->Per salvare sul DB una risposta

4) eliminarisposta($IdRisposta)->Per eliminare dal DB una risposta

FPersistentManager analisi metodi presenti

DECIDO DI RINOMINARE tutto con “getObj” nelle classi già fatte per non ritoccare il persistent manager, ovvero un metodo standardizzato che ritorna l’oggetto dandogli come argomento il suo ID (PK)

Anche “saveObj” deve essere standardizzato

SUDDIVIDO I METODI IN MACROCATEGORIE

METODI DI BASE

1)getInstance()-> parte del pattern singleton FATTO

2) retrieveObj($class, $id)->metodo che fa la load di un oggetto mediante la sua classe ed il suo id

3) uploadObj($obj)->metodo simile a quello di prima per fare l’upload di un oggetto sul db

RETRIEVE

Paziente

1) retrievepazientefromemail($email)-> ritorna un paziente dandone in input la sua mail (è una credenziale di accesso univoca)

2) retrieveallpazienti()->ritorna un'array contenente tutti i pazienti della piattaforma (uso dell'admin)

3) retrieveappuntamentifrompaziente($IdPaziente)-> ritorna un'array contenente tutti gli appuntamenti della di un determinato paziente dando come argomento il suo Id

Medico

1) retrievemedicofromemail($email)-> ritorna un medico dando come argomento la sua mail (è una credenziale di accesso univoca)

2) retrieveallmedici()->ritorna un'array contenente tutti i medici della piattaforma (uso dell'admin)

3) retrievemedicipropic($mediciArray) -> ritorna un'array contenente le propic dei medici contenuti nell'array inserito come argomento

4) retrievemedicieimmagini($mediciInput)-> $mediciInput è un'array di medici fornito come argomento e ritorniamo un'array contenente il medico con la relativa IdImmagine di propic

5) retrivemediciattivifromtipologia($IdTipologia)-> ritorna un'array di medici non bannati della categoria inserita

6) retrievemedicifromtipologia($IdTipologia)-> ritorna un'array contenenti tutti i medici della tipologia inserita come argomento (uso dell'admin)

7) retireveaveragevalutazione($IdMedico)-> ritorna la valutazione media (data dalle recensioni) di un medico il cui Id è fornito come argomento

8) retrievenumerorecensionimedico($IdMedico)-> ritorna il numero di recensioni di un medico il cui id è fornito come argomento del metodo

9) retrieveagendamedico($IdMedico)-> AGENDA DEL MEDICO??? COMPLESSO metodo che ritorna l'agenda (appuntamenti non scaduti) di un medico il cui Id è inserito come argomento

ADMIN

1) retrieveamministratorefromemail($email)-> ritorna un amministratore dandone in input la sua mail (è una credenziale di accesso univoca)

2) retrieveallappuntamenti()-> ritorna un'array contenente tutti gli appuntamenti della piattaforma (uso dell'admin)

VERIFICHE

1) verificaemailpaziente($email)-> verifica se esiste un paziente con la mail data in input

2) verificaemailmedico($email)-> verifica se esiste un medico con la mail in input

3) validaimmagine($file)-> controllo sull'immagine, se non è corretta (formato e dimensione) ritorna errore

UPDATE/UPLOAD

caricaimmagine($file) ->Metodo generale che applica la validazione dell'immagine e crea l'oggetto immagine per poi metterlo nel db

Paziente

1) updateinfopaziente($paziente)-> Metodo per l'aggiornamento dei campi del paziente (residenza e numero di telefono) dando in input l'oggetto paziente

2) aggiornabanpaziente($paziente) -> metodo che aggiorna l'attributo "attivo" di un paziente per bannarlo o sbannarlo

3) updatemailpaziente($paziente) -> Metodo per aggiornare la mail di un paziente dando in input l'oggetto paziente

4) updatepasswordpaziente($paziente)-> Metodo per aggiornare la password di un paziente dando in input l'oggetto paziente

Medico

1) updateinfomedico($medico)-> Metodo per aggiornare costo e propic di un medico dando in input l'oggetto medico

2) aggiornabanmedico($medico)-> Metodo che aggiorna l'attributo "attivo" di un medico per bannarlo o sbannarlo dando in input l'oggetto medico

3) updatemailmedico($medico) -> Metodo per aggiornare la mail di un medico dando in input l'oggetto medico

4) updatepasswordmedico($medico)-> metodo per aggiornare la password di un medico dando in input l'oggetto medico

5) updatemedicopropic($medico)-> Metodo per aggiornare sul db il campo IdImmagine dando in pasto l'oggetto medico ((ATTENZIONE, DA TESTARE))

6) caricaimmaginemedico(EImmagine $immagine, EMedico $medico)-> Metodo che salva l'immagine nel db e come propic del medico (INSIEME) ((SOSTITUTO DEL PRECEDENTE))

7) updateimmaginereferto($referto)-> Metodo per aggiornare sul db il campo IdImmagine dando in pasto l'oggetto referto ((ATTENZIONE, DA TESTARE))

8) caricaimmaginereferto(EImmagine $immagine, EReferto $referto) )-> Metodo che salva l'immagine nel db e come contenuto del referto (INSIEME) ((SOSTITUTO DEL PRECEDENTE))

9) updateCommentBan($comment)-> per una eventuale modifica delle recensioni

((NON REALIZZATO ))

Magari è da implementare una visualizazzione delle recensioni fatte solo da utenti attivi e non disattivati DA FARE UN ENTITYMANAGER (QUERY COMPLESSA)

CANCELLAZIONI

1) liberafasciaoraria($IdFascia\_oraria)-> metodo che controlla se la fascia oraria è stata occupata da un appuntamento, altrimenti la rimuove, dando in input il suo id

2) cancellaRecensione($IdRecensione)-> Metodo che cancella una recensione dando in input il suo id

3) cancellaImmagine($IdImmagine) -> Metodo per cancellare un'immagine dal db dando il suo id in input

DA DEFINIRE

52) manageImages($uploadedImages, $post, $idUser)-> DA VEDERE MEGLIO IN SEGUITO(SERVONO I TMP