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

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

2) retriveObj($class, $id)->metodo non banale che fa la load si un oggetto dal suo id mediante la sua classe FATTO

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

FATTO

4) getListaMediciAttiviFromTipologia($idTipologia)->non dovrebbe servire a molto (medici non bannati)?

FATTO

5) retrivePazienteFromEmail($email)->restituisce il paziente dalla mail FATTO

6) retriveAmministratoreFromEmail($email)->restituisce l’amministratore dalla mail FATTO

retriveMedicoFromEmail($email)->restituisce il medico dalla mail FATTO

7) getaveragevalutazione($IdMedico)-> Mi implemento la valutazione media di un medico FATTO

10) liberafasciaoraria($IdFascia\_oraria)-> Per la cancellazione di una fascia oraria, con controllo se presente o meno un appuntamento nella fascia selezionata FATTO

11) cancellaRecensione($IdRecensione)-> Per la cancellazione di una recensione FATTO

13) cancellaImmagine($IdImmagine) -> PER LE IMMAGINI FATTO

15) caricaimmaginemedico(EImmagine $immagine, EMedico $medico)->per settare la propic del medico e salvarla nel Db FATTO

16) caricaimmaginereferto(EImmagine $immagine, EReferto $referto) )->per settare l’immagine del referto e salvarla nel Db FATTO

17) getmedicipropic($mediciArray) -> utilizzabile per un array di medici FATTO

19 getnumerorecensionimedico($IdMedico)->per ottenere il numero di recensioni di un medico FATTO

20) getFollowedUserList($idUser)-> AGENDA DEL MEDICO??? COMPLESSO

21) getappuntamentipaziente($IdPaziente)-> APPUNTAMENTI DI UN DATO PAZIENTE FATTO

22) getSearchedPost($keyword)-> serve un metodo che torna i medici con una tipologia

Metodo già presente nel caso di medici attivi ma implementato per il caso generale(magari serve all’admin) FATTO

24) getallappuntamenti()-> TUTTI GLI APPUNTAMENTI PER L’ADMIN FATTO

25) getallmedici()->TUTTI I MEDICI PER L’ADMIN FATTO

getallpazienti()->TUTTI I PAZIENTI PER L’ADMIN FATTO

29) updateinfomedico($medico)->per l’aggiornamento delle informazioni dei mediciFATTO

30) updateinfopaziente($paziente)-> per l’aggiornamento delle informazioni dei pazienti FATTO

31) aggiornabanmedico($medico)-> Per aggiornare il db con i ban dei medici FATTO

aggiornabanpaziente($paziente) -> Per aggiornare il db con i ban dei pazienti FATTO

32) updatemedicopropic($medico)-> da utilizzare per medici BISOGNA RITOCCARE ENTITY

updateimmaginereferto($referto)-> da utilizzare per referti BISOGNA RITOCCARE ENTITY ed anche FREFERTO

34) updatemailmedico($medico) -> per il cambio mail del medico FATTO

updatemailpaziente($paziente)-> per il cambio mail dei pazienti FATTO

35) updatepasswordmedico($medico)-> per l’aggiornamento della password FATTO

updatepasswordpaziente($paziente)-> per l’aggiornamento della password FATTO

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

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

38) verificaemailpaziente($email)->applicato alla registrazione per non avere mail duplicate nei pazienti FATTO

verificaemailmedico($email)-> applicato alla registrazione per non avere mail duplicate nei medici FATTO

40) loadUserPage($id)->vedere i propri appuntamenti

41) loadPostPerCategory($category)-> load dei medici dalla categoria

42) loadPostInExplore($idUser)-> INUTILE

43) loadPostInVisited($idPost)-> INUTILE

44) getFollowedList($idUser)->INUTILE

46 loadmedicieimmagine($mediciInput)-> BUONO PER CARICARE MEDICI E PROPIC MA RISULTA NECESSARIO RITOCCARE EMedico per il campo immagine ed i metodi relativi

49) loadCommentsAndUsersPic($idPost)-> per le recensioni di un medico

50) getLikesPage($idPost)-> boh inutile

51) getPostAndUser($idPost)->no

52) manageImages($uploadedImages, $post, $idUser)-> DA VEDERE MEGLIO

53) uploadImage($file)->COMODO

54) validateImage($file)-> SERVE