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

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

4) salvapaziente ($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) getadminfromid($id)->Per ottenere un admin utilizzando il suo id

3) salvaamministratore($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) getimmaginefromid($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) salvaimmagine($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) gettipologiafromid($IdTipologia)->Per ottenere una tipologia utilizzando il suo id

3) salvatipologia($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) getmedicofromid($id)->Per ottenere un medico utilizzando il suo id

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

4) savemedico($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) getcalendariofromid($IdCalendario)->Per ottenere un calendario utilizzando il suo id

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

4) savecalendario($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) getfasciaorariafromid($IdFascia\_oraria)->Per ottenere una fascia oraria utilizzando il suo id

3) salvafasciaoraria($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) getappuntamentofromid($IdAppuntamento)->Per ottenere un appuntamento utilizzando il suo id

3) salvaappuntamento($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) getrefertofromid($IdReferto)->Per ottenere un referto utilizzando il suo id

3) salvareferto($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

1)getInstance()-> parte del pattern singleton

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

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

4) getCommentList($idPost)->non dovrebbe servire a molto (medici non bannati)?

5) retriveUserOnUsername($username)->va modificato per usare l’email

6) retriveModOnUsername($username)->va modificato per gli admin ed i medici

7) getLikeNumber($idPost)-> inutile

8) retriveLike($idUser, $idPost) ->inutile

9) retriveFollow($idUser, $idFollowed)-> inutile

10) deletePost($idPost, $idUser)-> da riadattare per fasce orarie ed appuntamenti?

11) deleteLike($idLike, $idUser)-> da riadattare per le recensioni?

12) deleteRelatedReports($id, $field = null)-> BOH

13) deleteImage($idImage) -> PER LE IMMAGINI

14) deleteImage($idImage)-> boh

15) uploadImagePost(EImage $image, EPost $post)->DA MODIFICARE PER L’AGGIUNTA DELLE PROPIC DEI MEDICI E DELLE IMMAGINI AI REFERTI

16) getLikesUserOfAPost($idPost)->inutile

17) getUsersPofilePic($userArray) -> utilizzabile per un array di medici

18) getUsersPofilePic($userArray)-> inutile

19) getFollowerNumb($idUser)->inutile

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

21) getFollowerUserList($idUser)-> inutile?

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

23) getSearchedUsers($keyword)->INUTILE

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

25) getReportedComment()->SIMILE AL PRECEDENTE

26) loadHomePage($id)-> INUTILE

27) findCommon($array1, $array2)-> INUTILE

28) loadVip()-> INUTILE

29) updateUserInfo($user)->DA ritoccare per l’aggiornamento delle informazioni

30) updateUserVip($user)-> non serve a molto (FORSE PER I BAN)

31) updateUserBan($user)-> QUESTO è DA IMPLEMENTARE BENE PER MEDICI E PAZ

32) updateUserIdImage($user)-> da utilizzare per medici e referti

33) updateUserWarnings($user)->inutile

34) updateUserUsername($user)-> per il cambio mail

35) updateUserPassword($user)-> per l’aggiornamento della password

36) updatePostBan($post)->inutile

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

38) verifyUserEmail($email)->corretto, va solo ritoccato ma da tenere

39) verifyUserUsername($username)-> inutile

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

45) getFollowerList($idUser)->per i medici dalla tipologia

46) loadUsersAndImage($userInput)-> STESSA ROBA

47) loadPostsAndLikes($postArray)->INUTILE

48) loadFollLikeNumb($post)->no

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