**-----Class LoginForm-----**

/\*\*\* Submit login informations\* @param username a string representation of the username that users enter  
\* @param password a string representation of the password that users enter  
\* @throws IllegalArgumentException if the parameter is of wrong type  
\*/  
public static void submitLoginInfo(String username, String password)  
throws IllegalArgumentException {

LoginController.attemptToLogin(username, password);

}

/\*\*  
\* Notifies success/fail to user  
\* @param message a string representation of the message to be notified to the user.  
\*/  
public static void notifyGuest(String message)  
throws IOException{

notify message to user;

}

**-----Class LoginController-----**

/\*\*  
\* Receive string username from the view, check if that string is empty or not  
\* @param username a string representation of the username that users enter  
\* @return return true if username is a empty string, otherwise return false.  
\*/  
public Boolean checkIfUsernameEmpty(String username){

if username is empty

return true

else

return false

}

/\*\*\* Receive string password from the view, check if that string is empty or not  
\* @param password a string representation of the password that users enter  
\* @return return true if password is a empty string, otherwise return false.  
\*/  
public Boolean checkIfPasswordEmpty(String password){

if password is empty

return true

else

return false

}

/\*\*  
\* Retrieve the User object with the username that user provide from the database and check if the password is correct  
\* @param username a string representation of the username that users enter  
\* @param password a string representation of the password that users enter  
\* @return return true if user is valid, otherwise return false  
\* @throws SQLException if the connection encounters an error during an interaction with a data source  
\*/  
public Boolean checkIfUserValid(String username, String password)  
throws SQLException {

User currentUser = find a user with username from database

if currentUser == null

return false;

if currentUser.getPassword() != password

return false;

else

return true;

}

/\*\*\* If one of the two inputs (or both) is empty or invalid username/password, assign the error message to variable notifiyMessage. If the user manages to log in successfully, assign the success message to variable notifyMessage. Then notify user the message\* @param username a string representation of the username that users enter  
\* @param password a string representation of the password that users enter  
\*/  
public static void attemptToLogin(String username, String password){

Boolean isUsernameEmpty = checkIfUsernameEmpty(username)

Boolean isPasswordEmpty = checkIfPasswordEmpty(password)

if (isUsernameEmpty == true) && (isPasswordEmpty == true)

notifyMessage = ‘Username and password must not be blank’

else if isUsernameEmpty == true

notifyMessage = ‘Username must not be blank’

else if isPasswordEmpty == true

notifyMessage = ‘Password must not be blank’

else if checkIfUserValid(username, password) != true

notifyMessage = ‘The username or password you entered is incorrect’

else {

notifyMessage = ‘Log in successfully’

start user session

}

notifyGuest();

}

/\*\*  
\* Notifies success/fail to user  
\* @param message a string representation of the message to be notified to the user.  
\*/  
public static void notifyGuest(){

send this.notifyMessage to LoginForm class;

}

**-----Class User-----**

/\*\*  
\* Get password of an user  
\* @return password of an user.  
\*/  
public String getPassword(){

return this.password;

}