Johanna Corsini

Valeriy Kustar

Amanda Pokorney

Final Project: Graphs and Test Cases

Project Page Class Methods:

**Edit Project Method:**

public string EditProject(string projectNum, string projectDesc, string projectDueDates, string projectPhase, string projectDeliverables, int projectHours, string projectStatus, int? eID, int flag)

1. if (projectNum == "")

2. return "please enter Project Number";

3. if (conn == null)

4. return "Not valid connection";

5. conn.Open();

6. if (flag == 1)

7. String upDateProject = ("UPDATE Project SET description = '" + projectDesc +

"', dueDate = '" + projectDueDates +

"', prjPhase = '" + projectPhase +

"', deliverables = '" + projectDeliverables +

"', hoursNeeded = " + projectHours +

", prjStatus = '" + projectStatus +

"' WHERE prjNo = '" + projectNum + "';");

MySqlCommand cmd = new MySqlCommand(upDateProject, conn);

cmd.ExecuteNonQuery();

return "Project Changed";

8. else if (flag == 0)

9. return "FLAG PASSED INCORRECT VALUE";

10. string prj = null;

String getPrjDup = "SELECT prjNo FROM Project WHERE prjNo = '" + projectNum + "';";

MySqlCommand cmd1 = new MySqlCommand(getPrjDup, conn);

MySqlDataReader reader = cmd1.ExecuteReader();

11. while (reader.Read())

12. prj = reader.GetString("prjNo");

13. reader.Close();

14. if (prj != null)

15. return "Duplicate Project";

16. String addProject = ("INSERT INTO Project (prjNo, prjLeader, description, prjPhase, dueDate, deliverables, hoursNeeded, prjStatus)" +

"VALUES ('" + projectNum + "', " + "'" + eID + "', '" + projectDesc + "', '" + projectPhase + "', '" + projectDueDates + "', '" + projectDeliverables + "', '" + projectHours + "', '" + projectStatus + "');");

MySqlCommand cmd = new MySqlCommand(addProject, conn);

cmd.ExecuteNonQuery();

return "Project Added";

GRAPH HERE

Edge Coverage:

**Assign Employee Method**

public string AssignEmployee(int hours, String firstName, String lastName)

1. if (conn == null)

2. return "Not valid connection";

3. conn.Open();

int empID = 0;

String getID = "SELECT employeeID FROM Employee E WHERE E.fname = '" + firstName + "' AND E.lname = '" + lastName + "';";

MySqlCommand cmd = new MySqlCommand(getID, this.conn);

MySqlDataReader reader = cmd.ExecuteReader();

4. while (reader.Read())

5. empID = reader.GetInt32("employeeID");

6. Console.WriteLine(empID);

conn.Close();

conn.Open();

int hoursAvail = 0;

String getHours = "SELECT hoursAvail FROM Employee E WHERE '" + empID + "' = E.employeeID;";

MySqlCommand cmd1 = new MySqlCommand(getHours, this.conn);

MySqlDataReader reader1 = cmd1.ExecuteReader();

7. while (reader1.Read())

8. hoursAvail = reader1.GetInt32("hoursAvail");

9. conn.Close();

10. if(hours <= hoursAvail)

11. return "Not enough hours available to be added to project.";

12. conn.Open();

//this ensures the program doesn't crash due to duplicate entries in the db

string dupId = null;

String getEmpDup = "SELECT employeeID FROM WorksOn WHERE employeeID = '" + empID + "' AND prjNo = '" + projectID + "';";

MySqlCommand cmddup = new MySqlCommand(getEmpDup, this.conn);

MySqlDataReader reader2 = cmddup.ExecuteReader();

13. while (reader2.Read())

14. dupId = reader2.GetString("employeeID");

15. conn.Close();

16. if (dupId != null)

17. return "Duplicate Employee on Project";

18. conn.Open();

String setMyID = "INSERT INTO WorksOn (employeeID, prjNo) VALUES (" + empID + ",'" + projectID + "');";

MySqlCommand cmd2 = new MySqlCommand(setMyID, this.conn);

Console.WriteLine(cmd2.ExecuteNonQuery());

conn.Close();

//update hours in employee table hoursAvail - hours

int totalHours = 0;

totalHours = hoursAvail - hours; //get new hours available for employee

conn.Open();

String setHours = "UPDATE WorksOn SET hours = '" + hours + "'WHERE employeeID = '" + empID + "';";

MySqlCommand cmd5 = new MySqlCommand(setHours, this.conn);

Console.WriteLine(cmd5.ExecuteNonQuery());

conn.Close();

conn.Open();

String setAvailHours = "UPDATE Employee SET hoursAvail = '" + totalHours + "'WHERE employeeID = '" + empID + "';";

MySqlCommand cmd6 = new MySqlCommand(setAvailHours, this.conn);

Console.WriteLine(cmd6.ExecuteNonQuery());

conn.Close();

return "Employee Added to project";

GRAPH HERE

Edge Coverage:

**Remove Employee Method:**

1. if (conn == null)

2. return "Not valid connection";

3. conn.Open();

int checkEmpID = 0;

String checkID = "SELECT prjLeader FROM Project P WHERE P.prjNo = '" + projectID + "' ;";

MySqlCommand cmd0 = new MySqlCommand(checkID, this.conn);

MySqlDataReader reader0 = cmd0.ExecuteReader();

4. while (reader0.Read())

5. checkEmpID = reader0.GetInt32("prjLeader");

6. conn.Close();

//Get ID with first name and last name

conn.Open();

int empID = 0;

String getID = "SELECT employeeID FROM Employee E WHERE E.fname = '" + firstName + "' AND E.lname = '" + lastName + "';";

MySqlCommand cmd = new MySqlCommand(getID, this.conn);

MySqlDataReader reader = cmd.ExecuteReader();

7. while (reader.Read())

8. empID = reader.GetInt32("employeeID");

9. conn.Close();

10. if(checkEmpID != empID)

11. return "Project Leader cannot be removed.";

12. conn.Open();

int hoursAssigned = 0;

String getHours = "SELECT hours FROM WorksOn E WHERE " + empID + " = E.employeeID;";

MySqlCommand cmd1 = new MySqlCommand(getHours, this.conn);

MySqlDataReader reader1 = cmd1.ExecuteReader();

13. while (reader1.Read())

14. hoursAssigned = reader1.GetInt32("hours");

15. conn.Close();

conn.Open();

int hoursAvail = 0;

String availGetHours = "SELECT hoursAvail FROM Employee E WHERE " + empID + " = E.employeeID;";

MySqlCommand cmd2 = new MySqlCommand(availGetHours, this.conn);

MySqlDataReader reader2 = cmd2.ExecuteReader();

16. while (reader2.Read())

17. hoursAvail = reader2.GetInt32("hoursAvail");

18. conn.Close();

conn.Open();

String removeID = "DELETE FROM WorksOn WHERE employeeID = '" + empID + "' ";

MySqlCommand cmd3 = new MySqlCommand(removeID, conn);

Console.WriteLine(cmd3.ExecuteNonQuery());

int totalHours = 0;

totalHours = hoursAvail + hoursAssigned;

String setHours = "UPDATE Employee SET hoursAvail = '" + totalHours + "'WHERE employeeID = '" + empID + "';";

MySqlCommand cmd4 = new MySqlCommand(setHours, this.conn);

Console.WriteLine(cmd4.ExecuteNonQuery());

conn.Close();

return "Employee removed from project";

GRAPH HERE

Edge Coverage:

Profile Page Class Methods:

**EditMyHours Method:**

public string EditMyHours(int emphours, int? eID)

1. if (conn == null)

2. return "Not valid connection";

3. if (empHours >= 0)

4. conn.Open();

MySqlCommand cmd = new MySqlCommand("UPDATE Employee SET hoursAvail = '" + empHours + "'WHERE employeeID = '" + this.eID + "';", conn);

Console.WriteLine(cmd.ExecuteNonQuery());

conn.Close();

return "Hours Updated";

5. return "Invalid Input";

GRAPH HERE

Edge Coverage:

**EditVacation Method:**

public string EditVacation(string startDate, string endDate, int? eID)

1. if (conn == null)

2. return "Not valid connection";

3. conn.Open();

string dupId = null;

String getVacDup = "SELECT employeeID FROM Vacation WHERE employeeID = '" + eID + "' AND startDate = '" + startDate + "';";

MySqlCommand cmd1 = new MySqlCommand(getVacDup, conn);

MySqlDataReader reader = cmd1.ExecuteReader();

4. while (reader.Read())

5. dupId = reader.GetString("employeeID");

6. if (dupId != null)

7. conn.Close();

return "Duplicate Vacation";

8. MySqlCommand cmd = new MySqlCommand("INSERT into Vacation VALUES(" + eID + ", '" + startDate + "','" + endDate + "');", conn);

Console.WriteLine(cmd.ExecuteNonQuery());

conn.Close();

return "Vacation Days Created";

GRAPH HERE

Edge Coverage:

**GetAdmin method:**

public Boolean getAdmin(int? eID)

1. Boolean isAdmin = false;

int result = 0;

2. if (conn == null)

3. return false;

4. conn.Open();

MySqlCommand cmd = new MySqlCommand("SELECT admin FROM Employee WHERE employeeID = '" + this.eID + "';", conn);

MySqlDataReader reader = cmd.ExecuteReader();

5. while (reader.Read())

6. result = reader.GetInt16("admin");

7. conn.Close();

8. if(result == 1)

9. isAdmin = true;

10. return isAdmin;

GRAPH HERE

Edge Coverage:

Login Page Class Methods:

**Login Method:**

1. if(conn == null)

2. return -1;

3. conn.Open();

string getLogin = "SELECT employeeID FROM Employee WHERE email = '" + email + "' AND pssword = '" + password + "';";

MySqlCommand cmd = new MySqlCommand(getLogin, this.conn);

MySqlDataReader reader = cmd.ExecuteReader();

4. while (reader.Read())

5. eID = reader.GetInt16("employeeID");

6. if (eID == null)

7. Console.WriteLine ("Not valid login");

8. Console.WriteLine("Logged in");

9. conn.Close();

return eID;

GRAPH HERE

Edge Coverage:

**Change Password Method:**

public string ChangePassword(string newPassword)

1. if (newPassword.Length < 6)

2. return "Password must be at least 6 characters";

3. if(conn == null)

4. return "Not valid Connection";

5. conn.Open();

string setPassword = "UPDATE EMPLOYEE SET pssword = '" + newPassword + "' WHERE employeeID = " + eID + ";";

MySqlCommand cmd = new MySqlCommand(setPassword, conn);

Console.WriteLine(cmd.ExecuteNonQuery());

conn.Close();

return "Password updated";

GRAPH HERE

Edge Coverage:

Test Cases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Method | Test Path | Pre Conditions | Input | Expected Output | Result |
| 1 | EditProject | (1, 2) | in db: eID = 1 | “”, ””, ””, ””, ”” , 0, ””, 1, 0 | "please enter Project Number" |  |
| 2 | EditProject | (1, 3, 4, 5, 6, 8) | in db: eID = 1, projectNum = 1 | “1”, ””, ””, ””, ”” , 0, ””, 1, 1 | "Project Changed" |  |
| 3 | EditProject | (1, 3, 4, 5, 6, 7) | Connection to db invalid | “1”, ””, ””, ””, ”” , 0, ””, 1, 1 | Exception ex |  |
| 4 | EditProject | (1, 3, 4, 9, 10) | in db: eID = 1, | “2”, ””, ””, ””, ”” , 0, ””, 1, -1 | "FLAG PASSED INCORRECT VALUE" |  |
| 5 | EditProject | (1, 3, 4, 9, 11, 12, 13) | Connection to db invalid | “2”, ””, ””, ””, ”” , 0, ””, 1, 0 | Exception ex |  |
| 6 | EditProject | (1, 3, 4, 9, 11, 12, 14, 15, 14, 16, 17, 18) | in db: eID = 1, projectNum = 1 | “1”, ””, ””, ””, ”” , 0, ””, 1, 0 | "Duplicate Project" |  |
| 7 | EditProject | (1, 3, 4, 9, 11, 12, 14, 15, 14, 16, 17, 19, 20, 22) | in db: no projectnum = 3, eID = 1 | “3”, ””, ””, ””, ”” , 0, ””, 1, 0 | "Project Added" |  |
| 8 | EditMyHours | (1, 2, 3) | in db: eID = 1 | “eight”, 1 | Exception ex |  |
| 9 | EditMyHours | (1, 2, 4, 5) | in db: eID = 1 | “-12”, 1 | "Invalid Input" |  |
|  | EditMyHours | (1, 2, 4, 6, 7, 8) | Connection to db invalid | “1”, 1 | Exception ex |  |
|  | EditMyHours | (1, 2, 4, 6, 7, 9) | in db: eID = 1 | “2”, 1 | "Hours Updated" |  |
|  | EditVacation | (1, 2, 4, 5, 4, 6, 7) | in db: 1, 1/1/20, 1/3/20 | “1/1/20”, “1/2/20”, 1 | “Duplicate Vacation” |  |
|  | EditVacation | (1, 2, 4, 5, 4, 6, 8, 10) | in db: eID = 1 | “2/1/20”, ”2/2/20”, 1 | “Vacation Days Created” |  |
|  | EditVacation | (1, 2, 3) | Connection to db invalid | “2/1/20”, ”2/2/20”, 1 | Exception ex |  |
|  | GetAdmin | (1, 2, 3) | Connection to db invalid | 1 | Exception ex |  |
|  | GetAdmin | (1, 2, 4, 5, 4, 6, 7, 8, 9) | in db: eID = 2, admin = 1 | 2 | true |  |
|  | GetAdmin | (1, 2, 4, 5, 4, 6, 7, 9) | in db: eID = 1, admin = 0 | 1 | false |  |
|  | Login | (1, 2, 3) | Connection to db invalid | “”, ”” | Exception ex |  |
|  | Login | (1, 2, 4, 5, 4, 6, 7, 9) | not in db | “sevenwonders@yahoo.com”, “password” | null |  |
|  | Login | (1, 2, 4, 5, 4, 6, 8, 9) | in db: email = testing@yahoo.com, pssword = pword, eID = 1 | “testing@yahoo.com”, “pword” | 1 |  |