**Introduction to Database Programming**

**MarioCraft Phase 3 Personal Report**

Dylan Chan (T00173201)

**Table of Contents**

[Introduction 2](#_Toc468895011)

[Using GitHub for version control 2](#_Toc468895012)

[Using Adobe Photoshop CS6 to create images 2](#_Toc468895013)

[Register form 3](#_Toc468895014)

[Login form 8](#_Toc468895015)

[GameUser Menu 13](#_Toc468895016)

[Conclusion 15](#_Toc468895017)

[References 15](#_Toc468895018)

# Introduction

For this phase 3 of Introduction to database programming is to implement tables, triggers , sequences and views in the database and use windows form to create our GUI. For database connection we used oracle through Amazon Web Services. For this project phase I used the entity Framework.

For my GUI, I created 3 windows form, Register Menu, Login Menu and GameUser Menu. All three GUI are interlinked, you can go back and forth between them with buttons. However to get into GameUser Menu you must successfully logged to access the menu. To register the user must input an email and password. To Login the user must input correct email and password. The GameUser Menu retrieves users email and password. By default the password input are shown “ \* ” when user inputs password however user can disable this feature. The GameUser Menu displays user details. I Used Visual Studio 2013 and C# to create this project.

# Using GitHub for version control

For my induvial phase 3 I decided to use GitHub for version control as I am familiar with it, as I have been using it for my other modules. It allows me to version control my project.

**GitHub Link** <https://github.com/DylanTLChan/DylanTChan_Phase3_MarioCraft.git>

# Using Adobe Photoshop CS6 to create images

I used Adobe Photoshop CS6 to create and design my 3 images for registration, login menu and app icon. Used ICO online converter to convert my app icon which I .jpg to .ico so that I can import it to visual studio to use it.

# Register form

The Register form is the first screen the user will see when the play the game. The User will be asked to enter an email and password to register. I used my trigger which calls a sequence that increments the Id, so each user will get a unique Id. When user has successfully inputted an email and password, they will be stored and saved into the database. There is also input validation in the textbox fields when the any of the textbox fields are empty it will prompt the user to input the reoccurred details into the textbox field.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DylanChan\_MarioCraft\_Phase3

{

public partial class frmRegisterMenu : Form

{

private frmLoginMenu loginMenu;

public frmRegisterMenu()

{

InitializeComponent();

FormBorderStyle = FormBorderStyle.FixedSingle;

}

public frmRegisterMenu(frmLoginMenu loginMenu)

{

InitializeComponent();

this.loginMenu = loginMenu;

}

private void frmRegisterMenu\_Load(object sender, EventArgs e)

{

//reference https://social.msdn.microsoft.com/Forums/vstudio/en-US/77a2ce66-65dd-4fb8-b252-7327b2e154be/hiding-the-password-in-a-textbox-with-c?forum=netfxbcl

//hides password when user inputs in the password field.

txtPassword.UseSystemPasswordChar = true;

}

private void btnReg\_Click(object sender, EventArgs e)

{

if(txtEmail.Text.Equals(""))

{

MessageBox.Show("Please Enter in a Email", "Field Empty", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtEmail.Focus();

return;

}

if (txtPassword.Text.Equals(""))

{

MessageBox.Show("Please Enter in a Password", "Field Empty", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtPassword.Focus();

return;

}

//Register new Users

using(var context = new Model1())

{

GAMEUSER newRegUser = new GAMEUSER()

{

USEREMAIL = txtEmail.Text,

USERPASSWORD = txtPassword.Text,

};

try

{

var result = context.GAMEUSERs.Add(newRegUser);

context.SaveChanges();

MessageBox.Show("Email: " + newRegUser.USEREMAIL + " and " + " Password: " + newRegUser.USERPASSWORD +

" \nYou have Successfully Registered","Registered ", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch(Exception ex)

{

MessageBox.Show(ex.GetBaseException().ToString(),"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

//Resets Textbox

txtEmail.Text = "";

txtPassword.Text = "";

}

private void btnLogin\_Click(object sender, EventArgs e)

{

frmLoginMenu frmNext = new frmLoginMenu();

this.Close();

frmNext.Show();

}

private void btnExit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void radBtnShow\_CheckedChanged(object sender, EventArgs e)

{

txtPassword.UseSystemPasswordChar = false;

}

private void btnRadHide\_CheckedChanged(object sender, EventArgs e)

{

txtPassword.UseSystemPasswordChar = true;

}

}

}

Successfully Registered New User

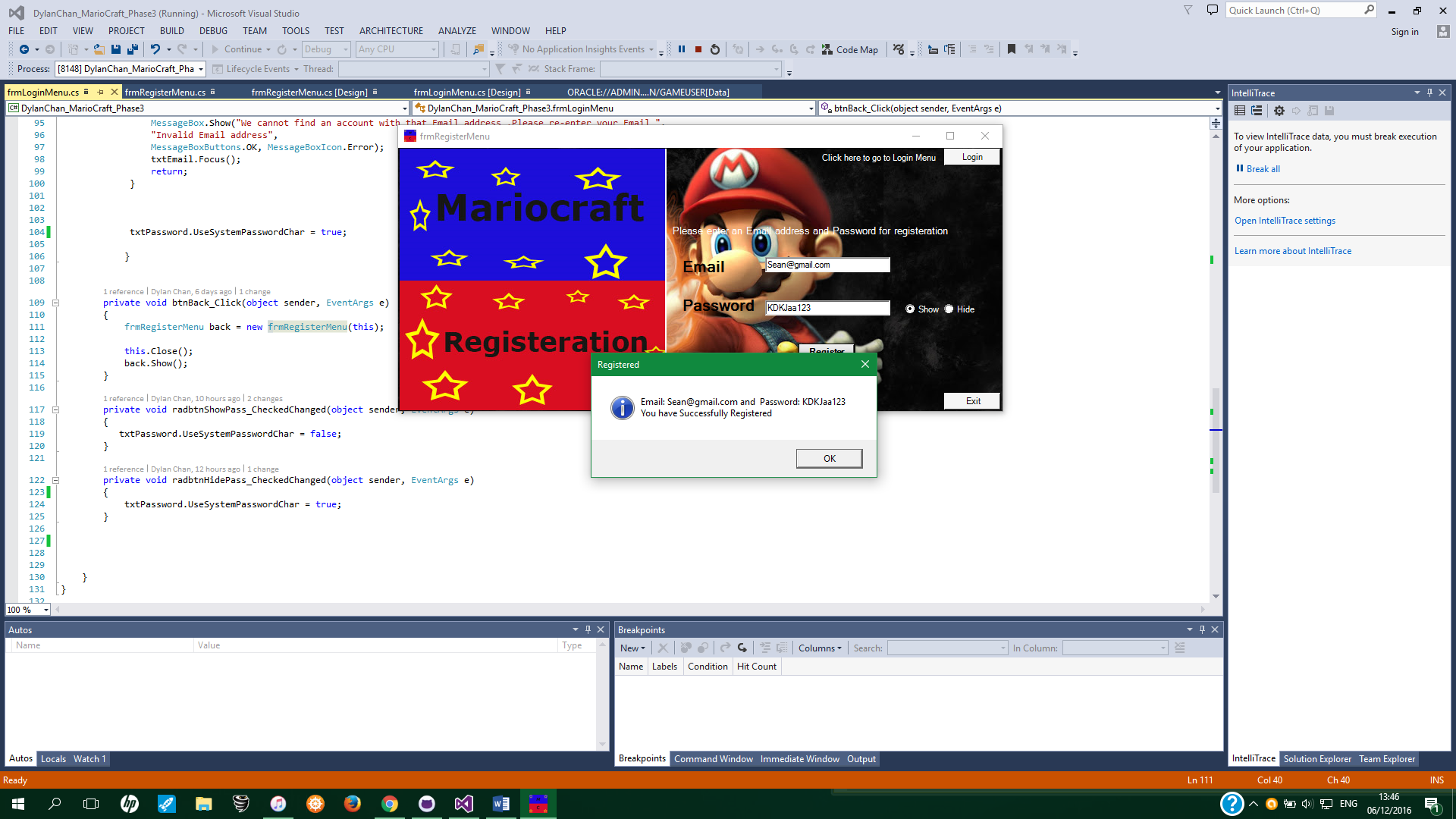
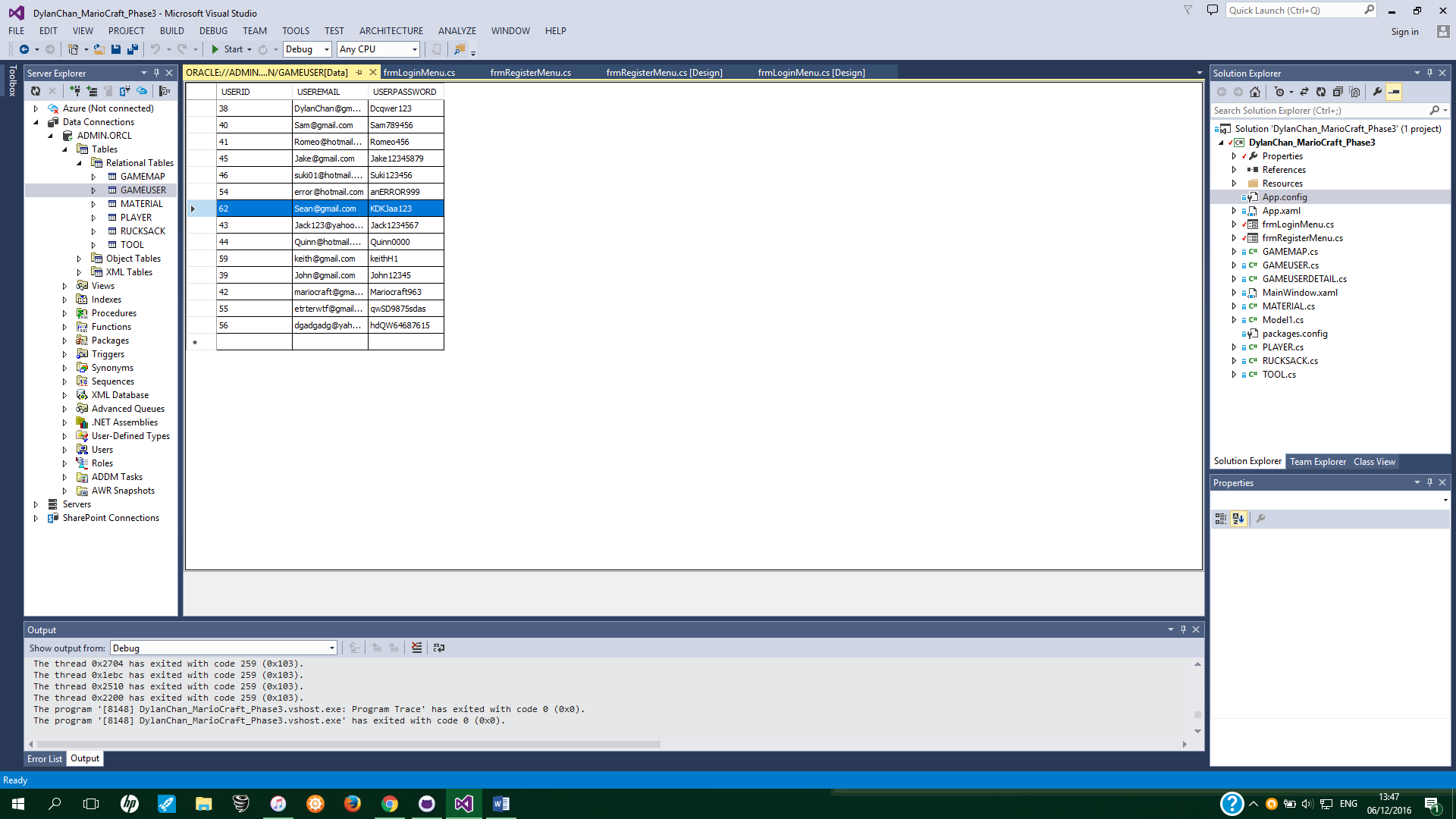
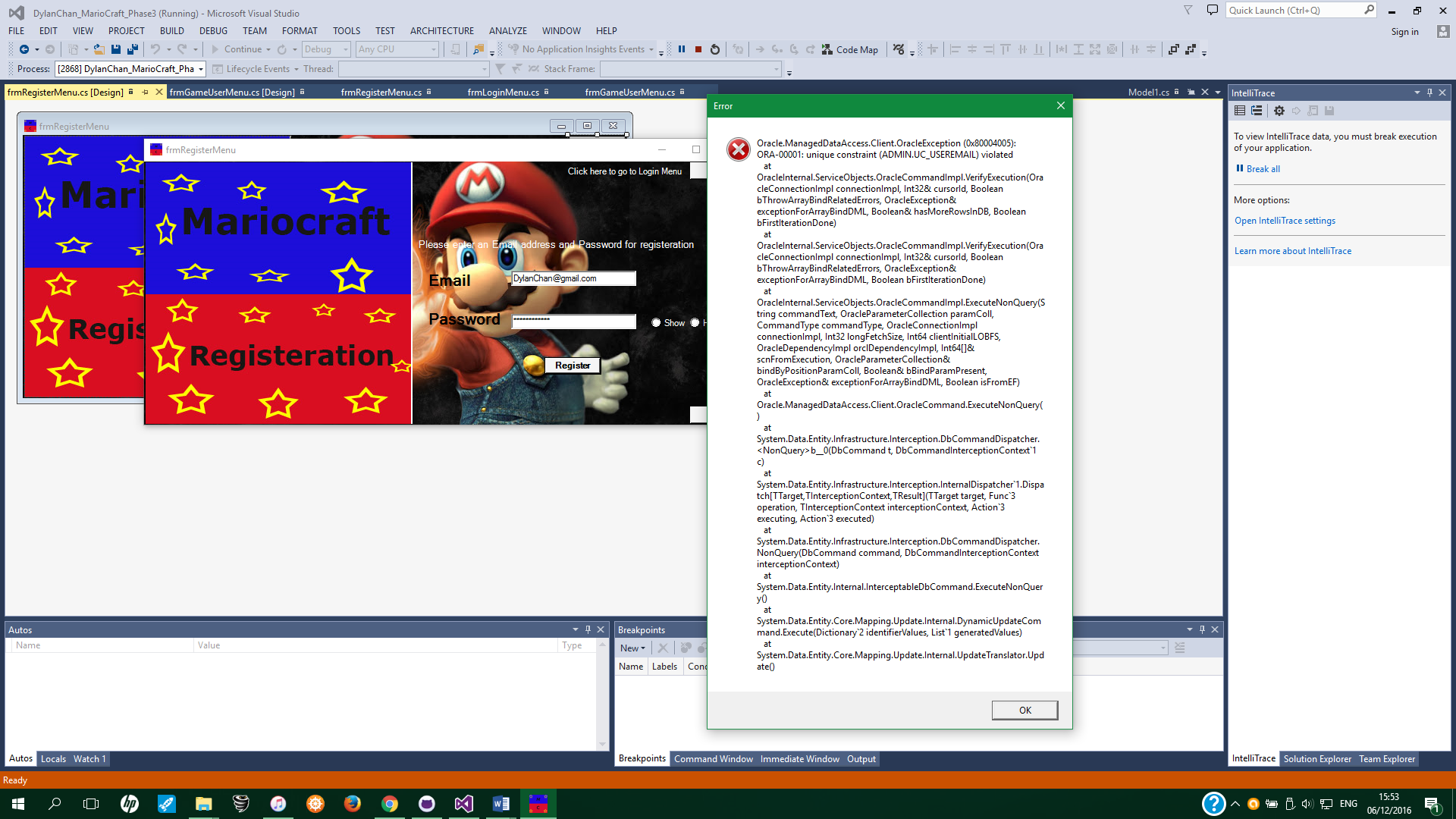


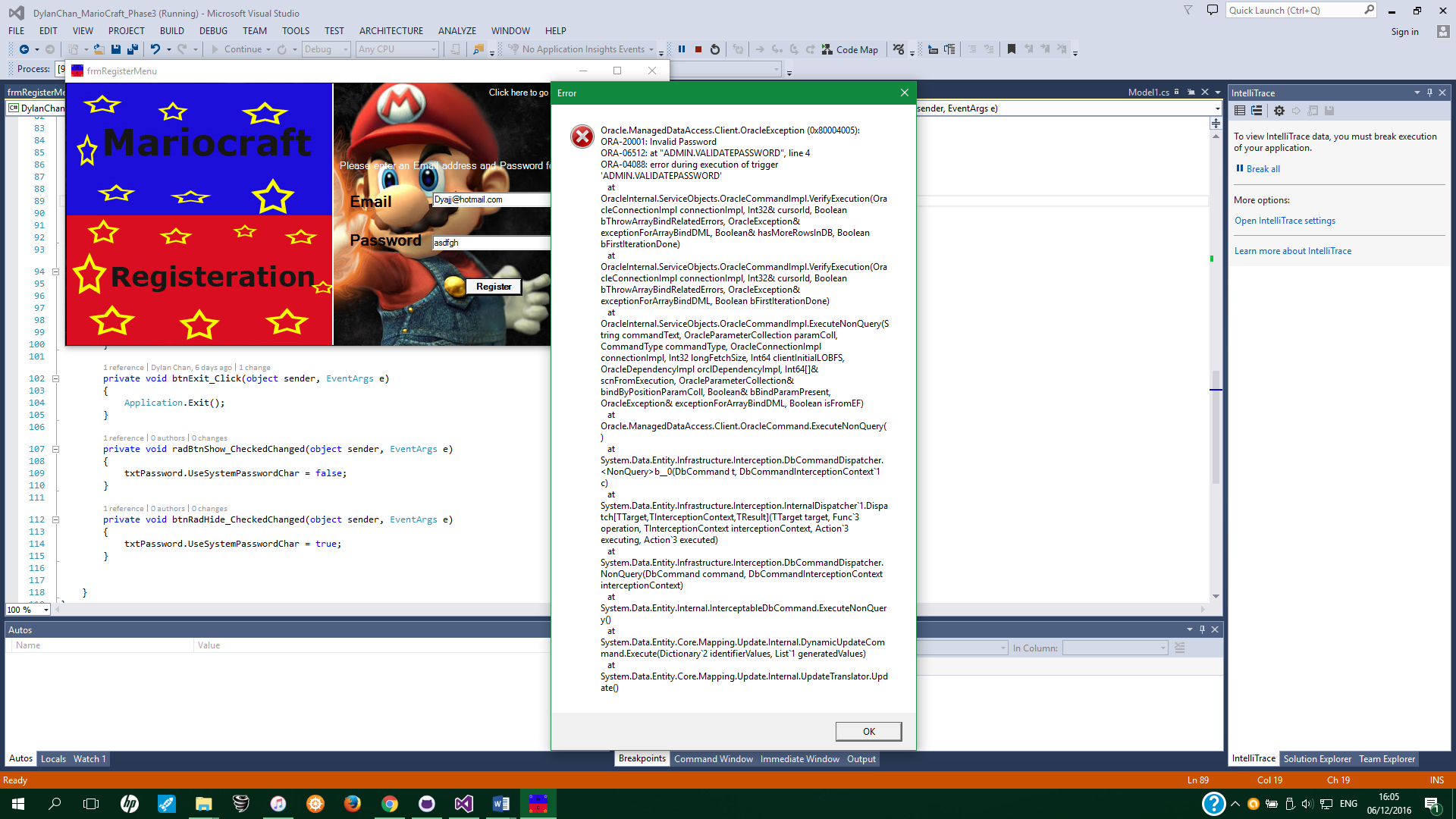
Table that contains newly registered user Sean@gmail.com and password: KDKJaa123



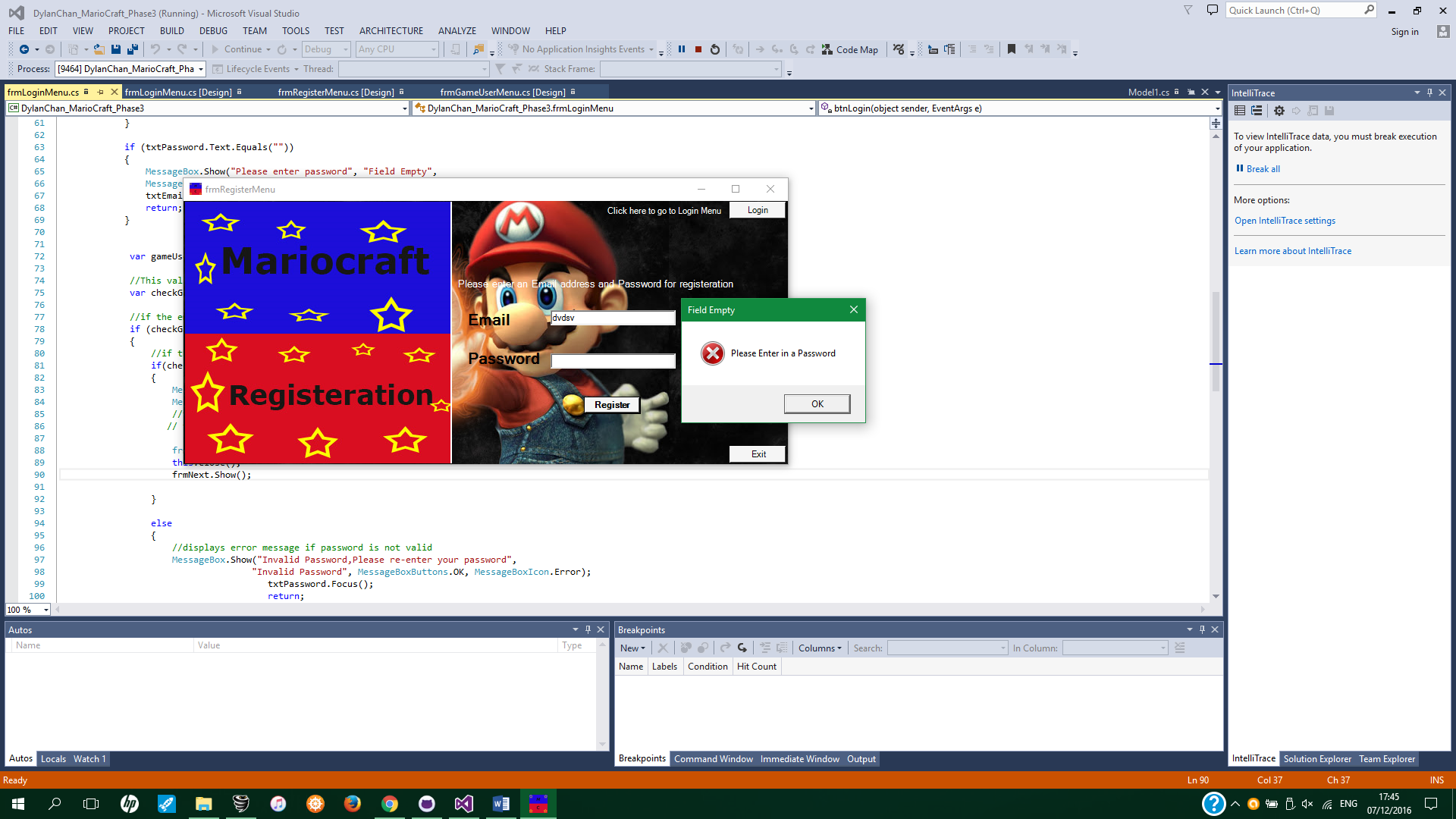
When enters email that already exist in the database an error message will show up.



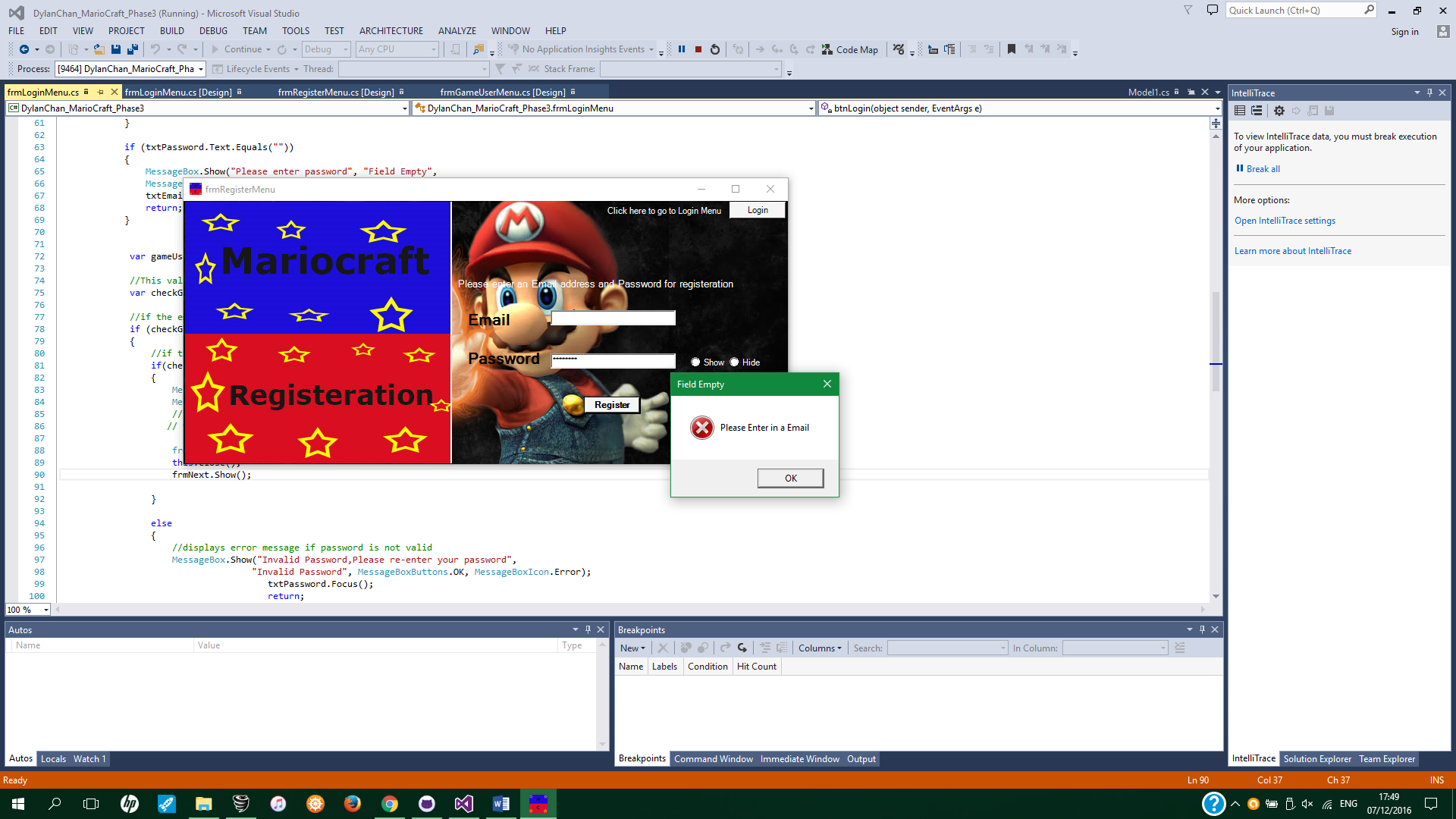
Invalid password entered error.



When password field empty, error message will show up



When Email field empty, error message will show up



# Login form

The Login menu is my second screen which can be accessed when you click on the login button in register menu. You can go back to register menu if user wish to go by clicking the back button. I have validation for when user inputs an invalid email and password. It will check the database if the email and password matches its record that it has, if they don’t it will display the appropriate error message to re-enter either email or password. Also the user cannot leave any of the textbox field empty if they do it will display an error message that will request the user to enter which ever field is empty. If the user successfully enters both valid email and password and clicks login, the user will automatically be brought to the GameUser menu. By default when the user enters the password, it will be hidden, however I have put in two radio button to be able to enable and disable the feature.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DylanChan\_MarioCraft\_Phase3

{

public partial class frmLoginMenu : Form

{

private frmRegisterMenu frmRegisterMenu;

Model1 context = new Model1();

private frmGameUserMenu frmGameUserMenu;

public frmLoginMenu()

{

InitializeComponent();

}

public frmLoginMenu(frmRegisterMenu frmRegisterMenu)

{

this.frmRegisterMenu = frmRegisterMenu;

FormBorderStyle = FormBorderStyle.FixedSingle;

}

public frmLoginMenu(frmGameUserMenu frmGameUserMenu)

{

this.frmGameUserMenu = frmGameUserMenu;

}

private void frmLoginMenu\_Load(object sender, EventArgs e)

{

//reference https://social.msdn.microsoft.com/Forums/vstudio/en-US/77a2ce66-65dd-4fb8-b252-7327b2e154be/hiding-the-password-in-a-textbox-with-c?forum=netfxbcl

//hides password when user inputs in the password field.

txtPassword.UseSystemPasswordChar = true;

}

private void btnExit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void btnLogin(object sender, EventArgs e)

{

if (txtEmail.Text.Equals(""))

{

MessageBox.Show("Please enter Email", "Field Empty",

MessageBoxButtons.OK, MessageBoxIcon.Error);

txtEmail.Focus();

return;

}

if (txtPassword.Text.Equals(""))

{

MessageBox.Show("Please enter password", "Field Empty",

MessageBoxButtons.OK, MessageBoxIcon.Error);

txtEmail.Focus();

return;

}

var gameUser = from u in context.GAMEUSERs where u.USEREMAIL == txtEmail.Text select u;

//This validates if gameuser's email is in the database

var checkGameUser = gameUser.FirstOrDefault(a => a.USEREMAIL.Equals(txtEmail.Text));

//if the email is not in the database

if (checkGameUser != null)

{

//if the password is in the database it will display successful dialog

if(checkGameUser.USERPASSWORD.Equals(txtPassword.Text))

{

MessageBox.Show("You have Successful logged in", "Login succesful",

MessageBoxButtons.OK, MessageBoxIcon.Information);

//txtEmail.Text = "";

// txtPassword.Text = "";

frmGameUserMenu frmNext = new frmGameUserMenu(txtEmail.Text);//email that is inputed by user

this.Close();

frmNext.Show();

}

else

{

//displays error message if password is not valid

MessageBox.Show("Invalid Password,Please re-enter your password",

"Invalid Password", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtPassword.Focus();

return;

}

}

else

{

//displays error message if Email is not valid

MessageBox.Show("We cannot find an account with that Email address ,Please re-enter your Email ",

"Invalid Email address",

MessageBoxButtons.OK, MessageBoxIcon.Error);

txtEmail.Focus();

return;

}

txtPassword.UseSystemPasswordChar = true;

}

private void btnBack\_Click(object sender, EventArgs e)

{

frmRegisterMenu back = new frmRegisterMenu(this);

this.Close();

back.Show();

}

private void radbtnShowPass\_CheckedChanged(object sender, EventArgs e)

{

txtPassword.UseSystemPasswordChar = false;

}

private void radbtnHidePass\_CheckedChanged(object sender, EventArgs e)

{

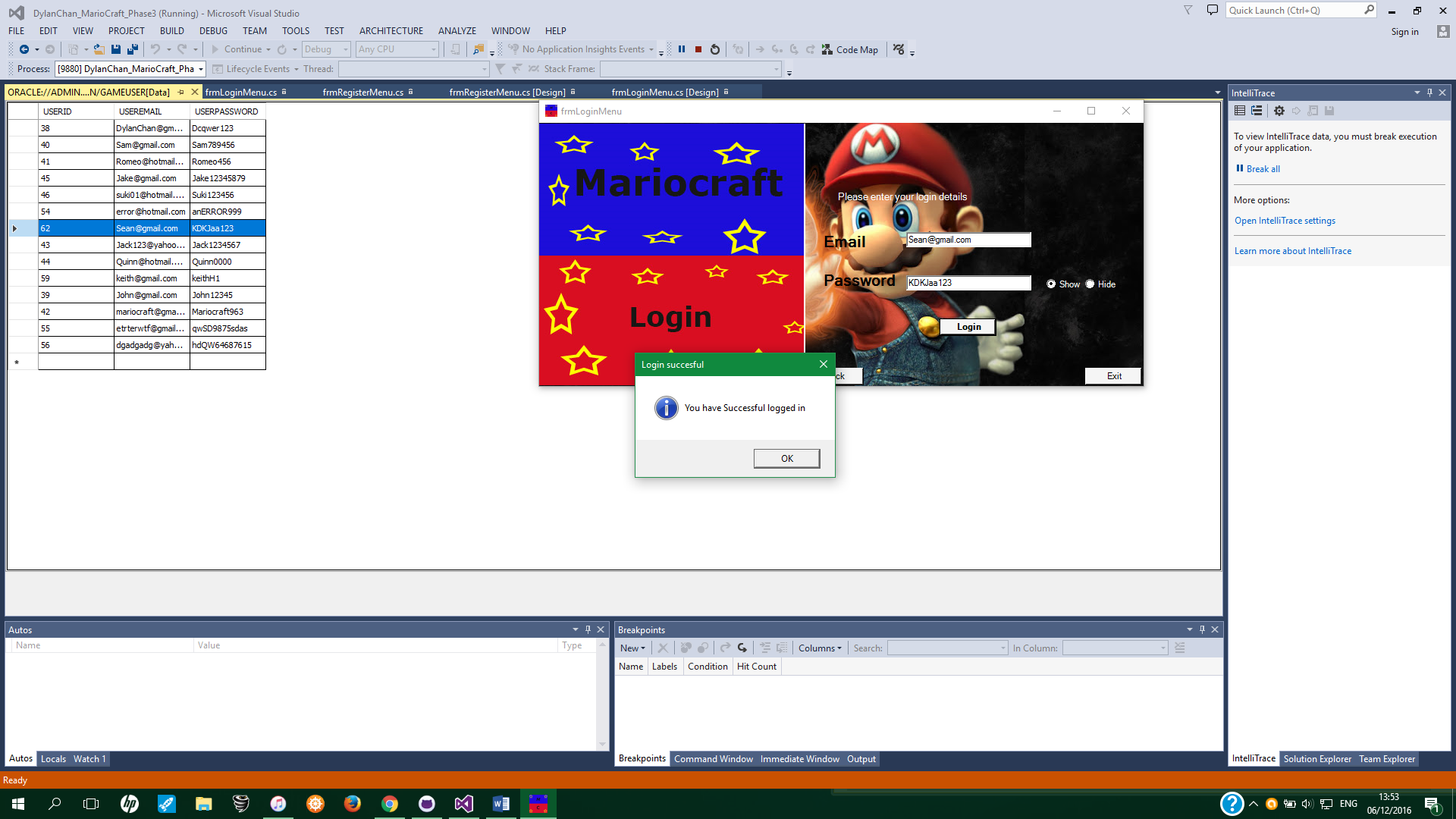
txtPassword.UseSystemPasswordChar = true;

}

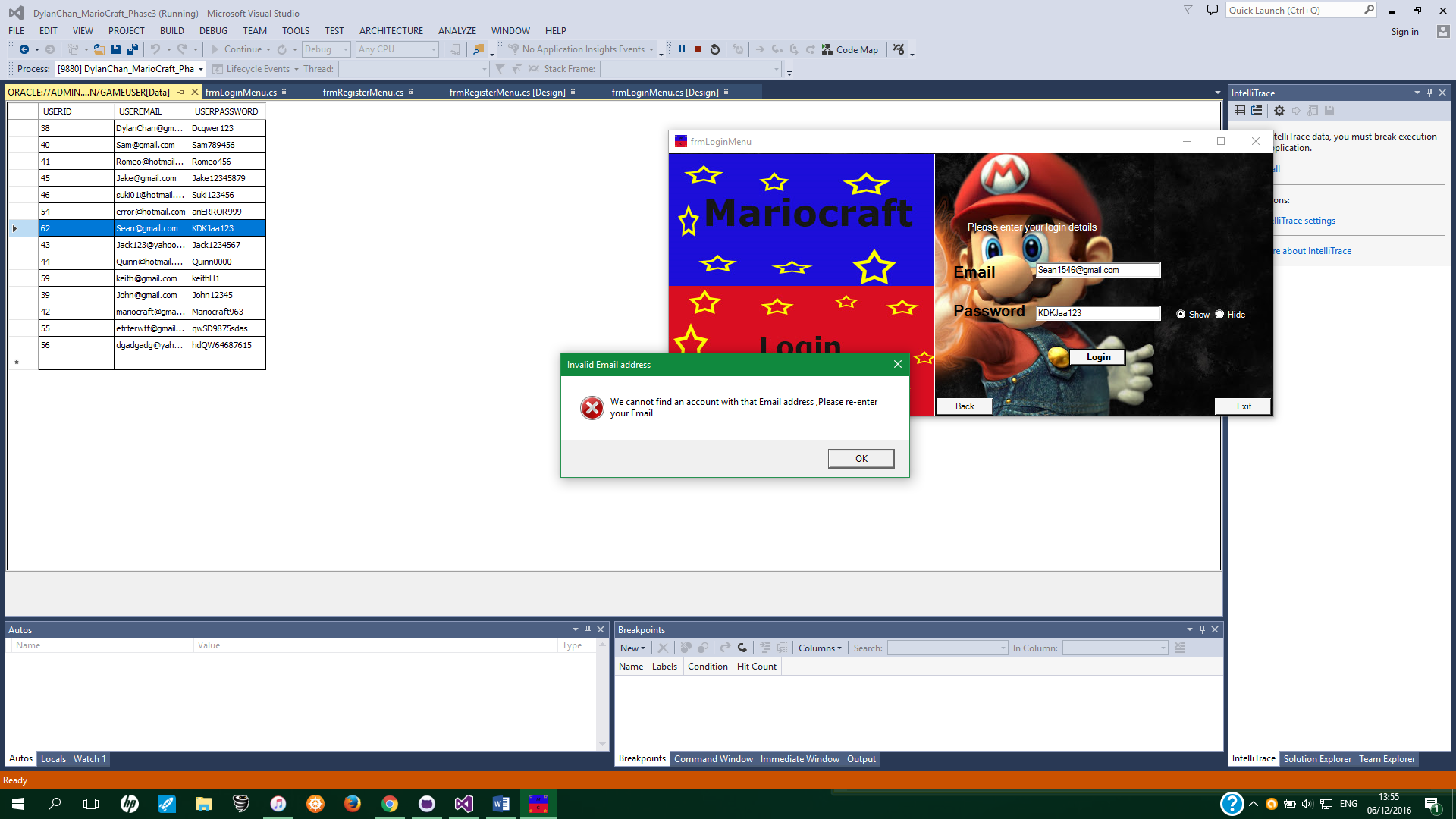
}

}

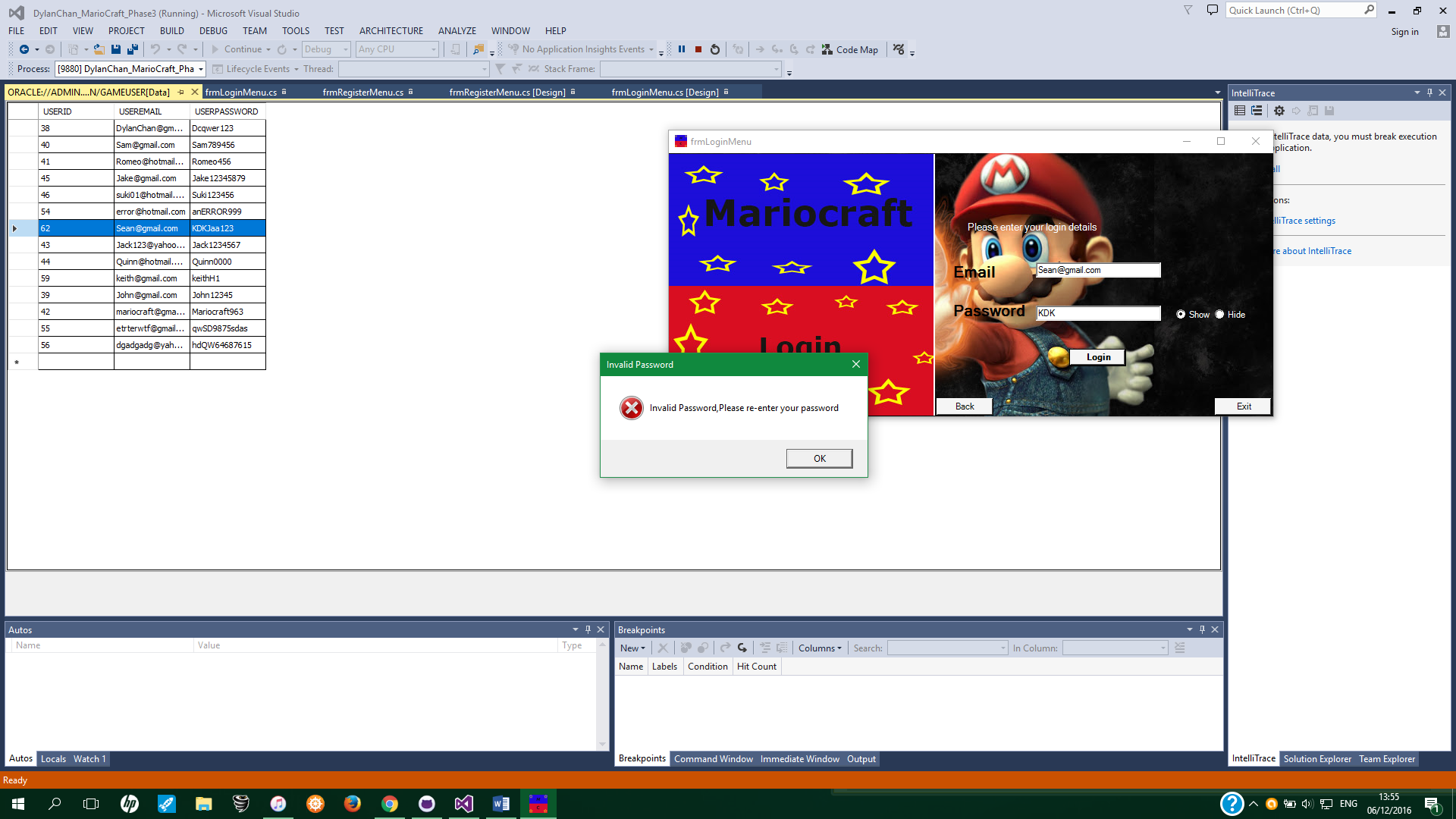
Successfully Logged in



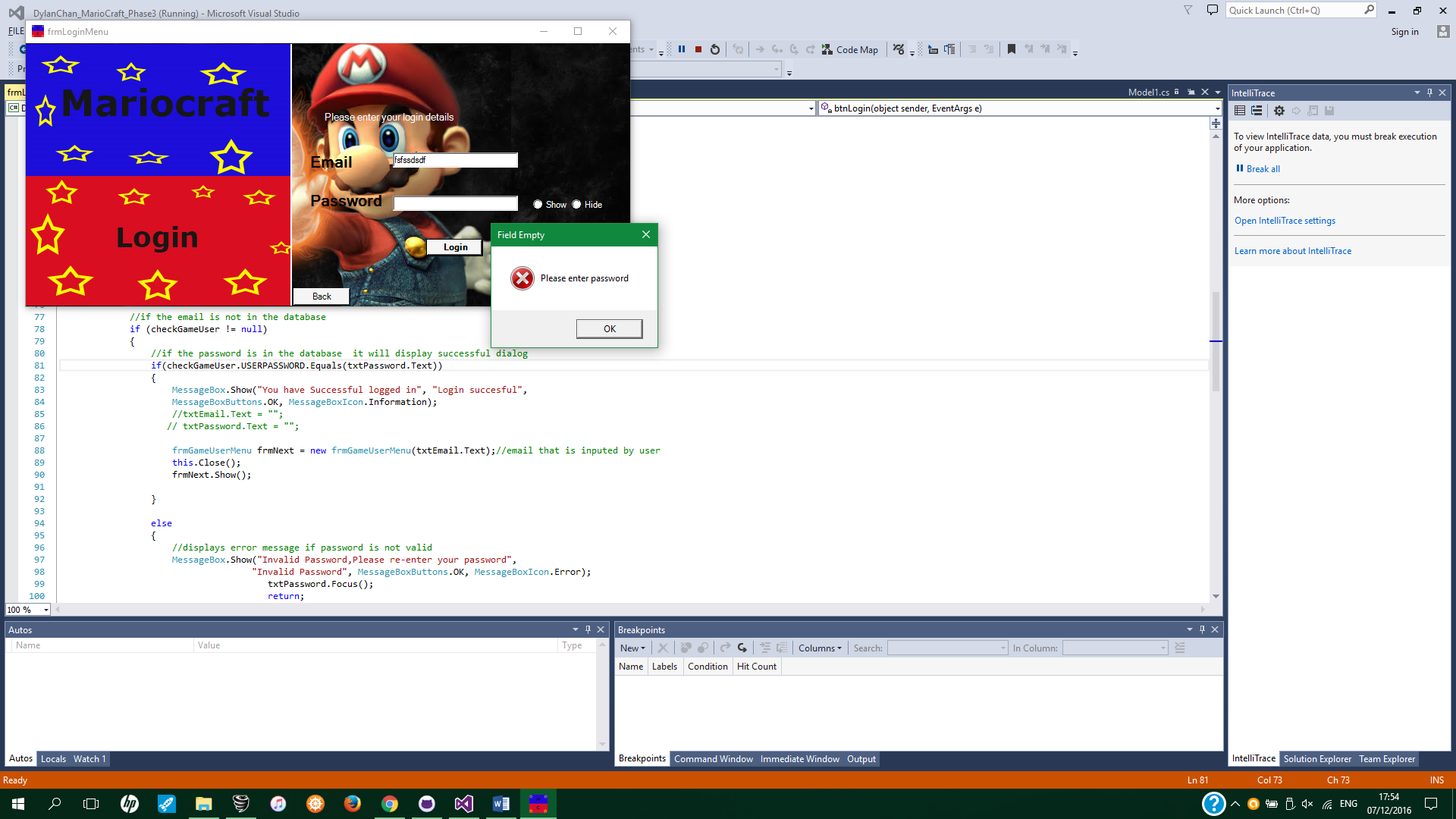
Invalid Email address entered.



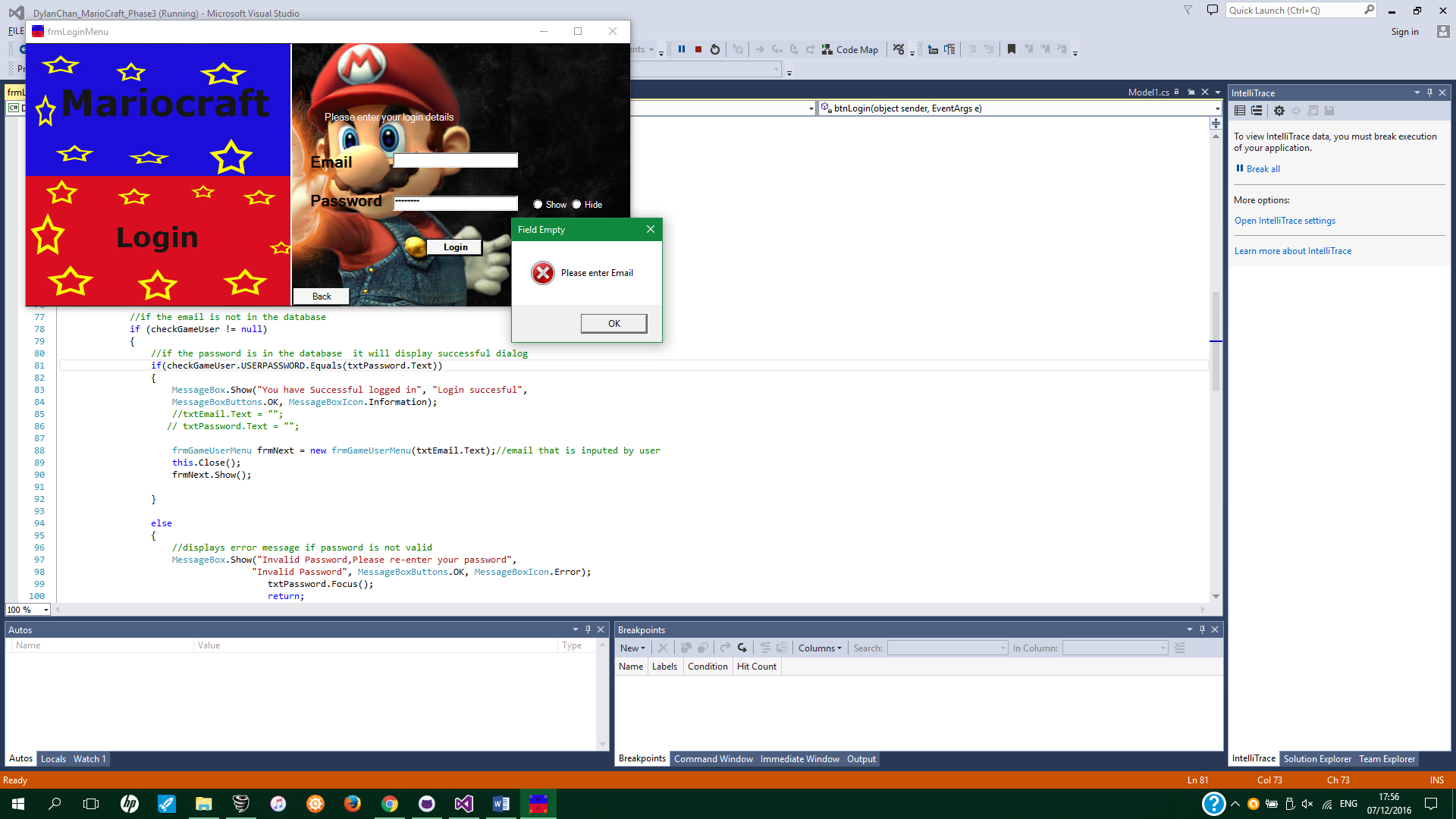
Invalid Password entered.



When password field empty, error message will show up.



When Email field empty, error message will show up.



# GameUser Menu

The GameUser Menu is the third screen which displays the details (UserId and User Email) of however successfully logs in. From the Login Menu there is this line of code “ frmGameUserMenu frmNext = new frmGameUserMenu(txtEmail.Text); ” When the user clicks login button in the login menu the Email information will be used to determine who the user is and correctly retrieve and displays the details in the datagrid view (this.gAMEUSERDETAILSTableAdapter.FillBy(dataSet1.GAMEUSERDETAILS,email);)

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DylanChan\_MarioCraft\_Phase3

{

public partial class frmGameUserMenu : Form

{

Model1 ctx = new Model1();

private frmLoginMenu frmLoginMenu;

private String email;

public frmGameUserMenu()

{

InitializeComponent();

}

public frmGameUserMenu(String email)

{

InitializeComponent();

this.email = email;

}

private void GameUserMenu\_Load(object sender, EventArgs e)

{

//This line of code loads data into the 'dataSet1.GAMEUSERDETAILS' table.

this.gAMEUSERDETAILSTableAdapter.FillBy(dataSet1.GAMEUSERDETAILS,email);

}

public frmGameUserMenu(frmLoginMenu frmLoginMenu)

{

this.frmLoginMenu = frmLoginMenu;

}

//Retrieve

/\*Model1 ctx = new Model1();

var retriveItem = from gameUser in ctx.RUCKSACKs

where gameUser.ITEMDESC == "Opal"

select gameUser;

this.txtItem.Text = retriveItem.First().ITEMDESC;\*/

private void btnExit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void txtLogOut\_Click(object sender, EventArgs e)

{

frmLoginMenu logout = new frmLoginMenu();

this.Close();

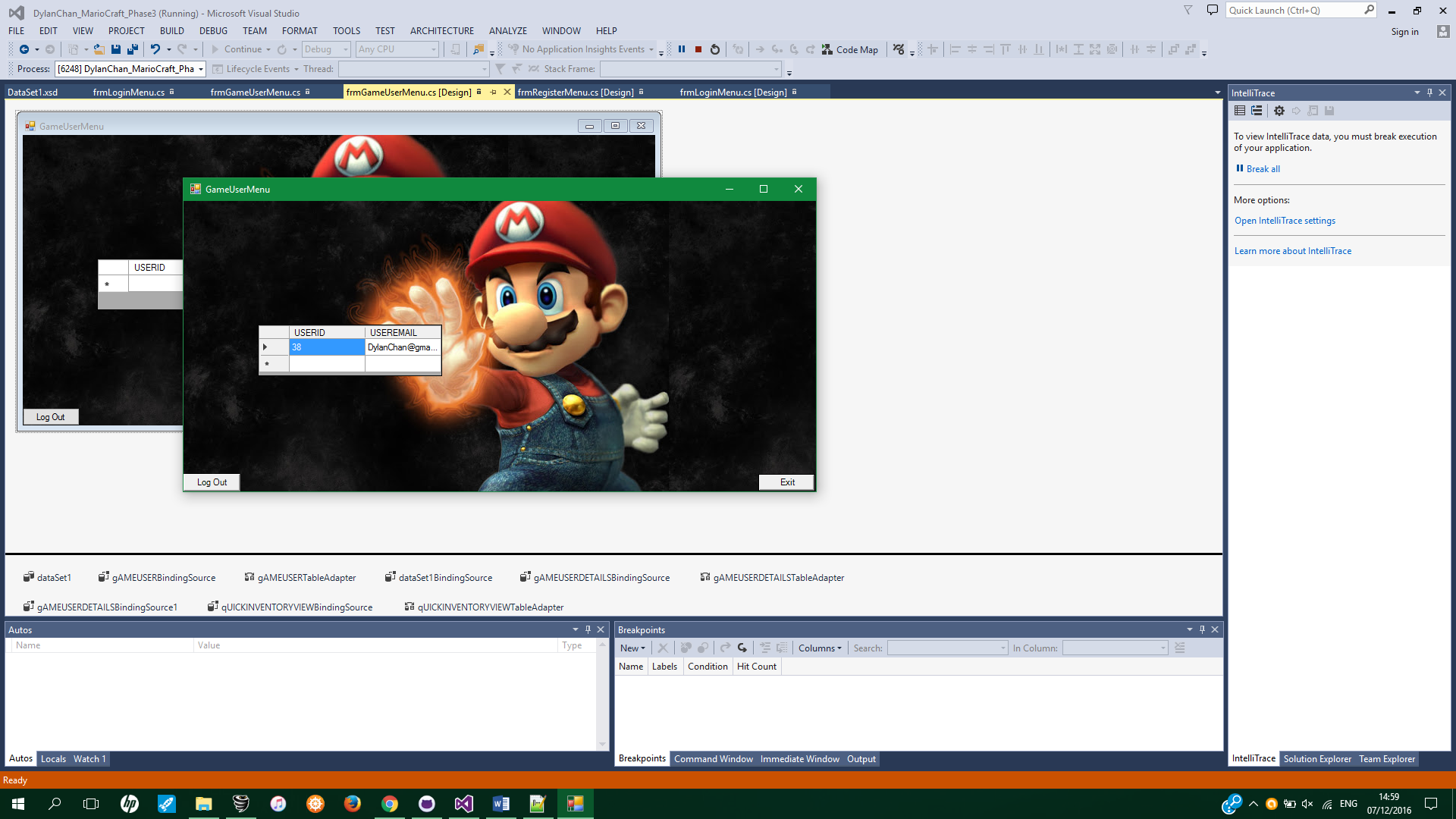
logout.Show();

}

}

}

User details displayed in a Datagrid view



# Conclusion

In conclusion, I successfully created register, login and gameuser menus. The register Menu successfully added emails and passwords to the database and implemented triggers and sequence. Login Menu successfully able read from database and validate if the details entered by the user is valid. Gameuser menu successfully retrieved data from database and displayed it to the user using a datagrid view. Using the Entity framework made coding the program much faster and less coding required, however I had difficulties setting up the Entity framework in Visual Studio, eventually I got the entity framework to work with the help of my team members. Overall I enjoyed this phase of the project as I can implement what I have learnt to the project.

# References

Background Image reference <http://www.taringa.net/posts/imagenes/15490026/Wallpapers-Mario-Bros-HD.html>

Code to hide password in text field <https://social.msdn.microsoft.com/Forums/vstudio/en-US/77a2ce66-65dd-4fb8-b252-7327b2e154be/hiding-the-password-in-a-textbox-with-c?forum=netfxbcl>