Deliverable 2

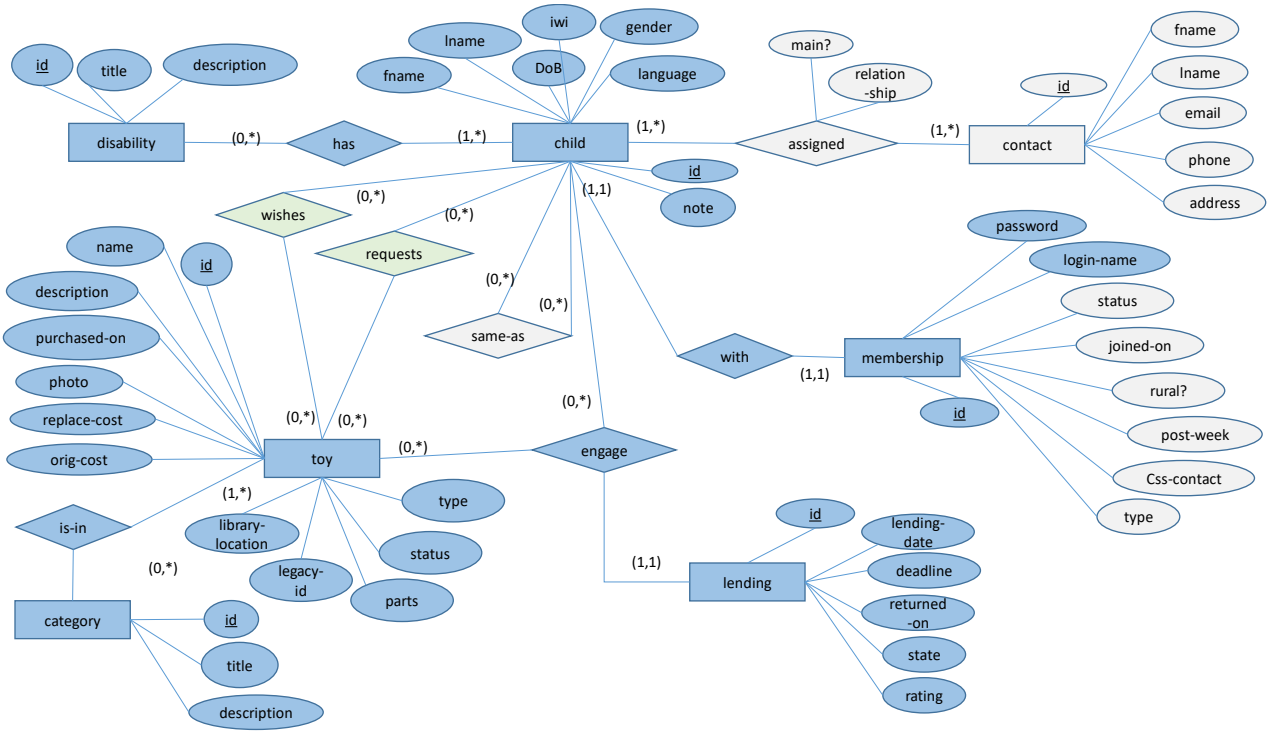
Name: Andre Ofsoski
ID: 1314669

Table of Contents

Introduction:

The purpose of this report is to provide information on the development of a database system for the Toy Library Hamilton. Firstly, in this report I will show the ER digram that you have specified for your needs and a recreational schema demonstrating how database layout. I will demonstrate an overview of the SQL code that creates the database and also the database with pre-filled tables as an example of what your data may look like inputted into the system. There will also be code demonstrated that allows the login of your clients and yourself, which allows you to login, register, browse users and a list of children registered. Finally, there will be screenshots of a walkthrough for you so that you may have a better idea on how to navigate and use the database.

ER Diagram Overview



Relational schema:

- disability (disabilityID, title, description)
- child (childID, note, language, gender, iwi, DoB, lname, fname, membershipID)
- membership (membershipID, password, login-name)
- lending (lendingID, lending-date, deadline, returned-on, state, rating, childID, toyID)
- toy (toyID, name, description, purchased-on, photo, replace-cost, orig-cost, library-location, legacy-id, parts, status, type)
- category (categoryID, title, description)
- has (disabilityID, childID)
- is-in (toyID, categoryID)

SQL Code

```
use ajo20_Deliverable2
create table Disability (
 disabilityID int not null primary key,
 title varChar(30) Unique,
 descriptionInjury varChar(60),
)

create table Membership(
membershipID int not null primary key,
memberPassword varChar(30),
loginName varChar(30) Unique,
)

create table Toy(
toyID int not null primary key,
toyName varChar(25),
toyDescription varChar(30),
purchasedOn varChar(15),
photo image,
replaceCost float,
origCost float,
libraryLocation varChar(15),
legacyID int,
parts varChar(25),
toyStatus varChar(15),
toyType varChar(15),
check (toyType= 'book/puzzle' or toyType = 'toy'),
check (toyStatus= 'available' or toyStatus= 'maintainance'),
check (libraryLocation= 'backroom' or libraryLocation= 'front room')
)

create table Child (
childID int not null primary key,
note varChar(50),
languageSpoken varChar(10),
gender varChar(10),
iwi varChar(30),
DoB varChar(20),
lname varChar(30),
fname varChar(30),
membershipID int,
foreign key (membershipID) references Membership
)

create table Lending(
lendingID int not null primary key,
lendingDate date,
deadline date,
returnedOn date,
stateOfToy varChar(30),
rating int,
childID int,
toyID int,
foreign key(childID) references Child,
foreign key(toyID) references Toy,
check (stateOfToy='broken' or stateOfToy='dirty' or stateOfToy='lost' or
stateOfToy='parts missing' or stateOfToy='fine'),
check (rating='1' or rating='2' or rating='3' or rating='4' or rating='5')
)
```

```
create table Category(
categoryID int not null primary key,
title varChar(30) Unique,
categoryDescription varChar(60)
)

create table Has(
disabilityID int not null,
childID int not null,
foreign key(disabilityID) references Disability,
foreign key(childID) references Child
)

create table IsIn(
toyID int not null,
categoryID int not null,
foreign key(toyID) references Toy,
foreign key(categoryID) references Category
)

INSERT INTO Category Values(001,'Active Play','Toy Description Here')
INSERT INTO Category Values(002,'Animals','Toy Description Here')
INSERT INTO Category Values(003,'Asking Questions','Toy Description Here')
INSERT INTO Category Values(004,'Babies','Toy Description Here')
INSERT INTO Category Values(005,'Balance','Toy Description Here')
INSERT INTO Category Values(006,'Books','Toy Description Here')
INSERT INTO Category Values(007,'Cars','Toy Description Here')
INSERT INTO Category Values(008,'CD music','Toy Description Here')
INSERT INTO Category Values(009,'CD story','Toy Description Here')
INSERT INTO Category Values(010,'CDs','Toy Description Here')
INSERT INTO Category Values(011,'Colour','Toy Description Here')
INSERT INTO Category Values(012,'Communcation','Toy Description Here')
INSERT INTO Category Values(013,'Construction','Toy Description Here')
INSERT INTO Category Values(014,'Coordination','Toy Description Here')
INSERT INTO Category Values(015,'Counting','Toy Description Here')
INSERT INTO Category Values(016,'Dancing','Toy Description Here')
INSERT INTO Category Values(017,'Dice','Toy Description Here')
INSERT INTO Category Values(018,'Dinosaurs','Toy Description Here')
INSERT INTO Category Values(019,'Dolls','Toy Description Here')
INSERT INTO Category Values(020,'Dolls house','Toy Description Here')
INSERT INTO Category Values(021,'Dress up','Toy Description Here')
INSERT INTO Category Values(022,'DVD','Toy Description Here')
INSERT INTO Category Values(023,'DVD - Film','Toy Description Here')
INSERT INTO Category Values(024,'DVD - music','Toy Description Here')
INSERT INTO Category Values(025,'Early child hood','Toy Description Here')
INSERT INTO Category Values(026,'Fine Motor','Toy Description Here')
INSERT INTO Category Values(027,'Fire engine','Toy Description Here')
INSERT INTO Category Values(028,'Firemen','Toy Description Here')
INSERT INTO Category Values(029,'Flash cards','Toy Description Here')
INSERT INTO Category Values(030,'Floor play','Toy Description Here')
INSERT INTO Category Values(031,'Floor puzzles','Toy Description Here')
INSERT INTO Category Values(032,'Games','Toy Description Here')
INSERT INTO Category Values(033,'Gardening','Toy Description Here')
INSERT INTO Category Values(034,'Gears','Toy Description Here')
INSERT INTO Category Values(035,'Gross Motor','Toy Description Here')
INSERT INTO Category Values(036,'Hammering','Toy Description Here')
INSERT INTO Category Values(037,'House','Toy Description Here')
INSERT INTO Category Values(038,'Imaginative play','Toy Description Here')
INSERT INTO Category Values(039,'Jumping','Toy Description Here')
INSERT INTO Category Values(040,'Kaleidoscopes','Toy Description Here')
```

```
INSERT INTO Category Values(041,'Kitchen','Toy Description Here')
INSERT INTO Category Values(042,'Kits','Toy Description Here')
INSERT INTO Category Values(043,'Light','Toy Description Here')
INSERT INTO Category Values(044,'Literacy','Toy Description Here')
INSERT INTO Category Values(045,'Loud sounds','Toy Description Here')
INSERT INTO Category Values(046,'Machines','Toy Description Here')
INSERT INTO Category Values(047,'Magnets','Toy Description Here')
INSERT INTO Category Values(048,'Marble runs','Toy Description Here')
INSERT INTO Category Values(049,'Maths','Toy Description Here')
INSERT INTO Category Values(050,'Memory','Toy Description Here')
INSERT INTO Category Values(051,'Music','Toy Description Here')
INSERT INTO Category Values(052,'Nature','Toy Description Here')
INSERT INTO Category Values(053,'Outdoor games','Toy Description Here')
INSERT INTO Category Values(054,'Phonetics','Toy Description Here')
INSERT INTO Category Values(055,'Police','Toy Description Here')
INSERT INTO Category Values(056,'Posting','Toy Description Here')
INSERT INTO Category Values(057,'Pretend play','Toy Description Here')
INSERT INTO Category Values(058,'Prisms','Toy Description Here')
INSERT INTO Category Values(059,'Puppets','Toy Description Here')
INSERT INTO Category Values(060,'Puzzles','Toy Description Here')
INSERT INTO Category Values(061,'Ride-on','Toy Description Here')
INSERT INTO Category Values(062,'Role play','Toy Description Here')
INSERT INTO Category Values(063,'Science','Toy Description Here')
INSERT INTO Category Values(064,'Sensory','Toy Description Here')
INSERT INTO Category Values(065,'Singing','Toy Description Here')
INSERT INTO Category Values(066,'Soft sounds','Toy Description Here')
INSERT INTO Category Values(067,'Space','Toy Description Here')
INSERT INTO Category Values(068,'Stacking','Toy Description Here')
INSERT INTO Category Values(069,'Taking turns','Toy Description Here')
INSERT INTO Category Values(070,'Tents','Toy Description Here')
INSERT INTO Category Values(071,'Texture','Toy Description Here')
INSERT INTO Category Values(072,'Threading','Toy Description Here')
INSERT INTO Category Values(073,'Trains','Toy Description Here')
INSERT INTO Category Values(074,'Trucks','Toy Description Here')
INSERT INTO Category Values(075,'Vehicles','Toy Description Here')
INSERT INTO Category Values(076,'Walking','Toy Description Here')
INSERT INTO Category Values(077,'Water play','Toy Description Here')
INSERT INTO Category Values(087,'Words','Toy Description Here')


INSERT INTO Disability Values(001,'ADHD','Attention Deficit Hyperactivity
Disorder')
INSERT INTO Disability Values(002,'ASD',' Autism spectrum disorder')
INSERT INTO Disability Values(003,'Asthma',' Disorder Description Here')
INSERT INTO Disability Values(004,'Blind',' Disorder Description Here')
INSERT INTO Disability Values(005,'Cerebral palsy ',' Disorder Description
Here')
INSERT INTO Disability Values(006,'Cystic fibrosis',' Disorder Description
Here')
INSERT INTO Disability Values(007,'Deaf',' Disorder Description Here')
INSERT INTO Disability Values(008,'Developmental delay',' Disorder
Description Here')
INSERT INTO Disability Values(009,'Downs syndrome',' Disorder Description
Here')
INSERT INTO Disability Values(010,'Dyslexia',' Disorder Description Here')
INSERT INTO Disability Values(011,'Epilepsy',' Disorder Description Here')
INSERT INTO Disability Values(012,'Traumatic brain injury',' Disorder
Description Here')
INSERT INTO Disability Values(013,'Intellectual disability',' Disorder
Description Here')
```

```
INSERT INTO Disability Values(014,'Multiple Sclerosis (MS)',' Disorder
Description Here')
INSERT INTO Disability Values(015,'Muscular Dystrophy',' Disorder
Description Here')
INSERT INTO Disability Values(016,'Paraplegia',' Disorder Description
Here')
INSERT INTO Disability Values(017,'Parkinsons disease',' Disorder
Description Here')
INSERT INTO Disability Values(018,'Physical disability',' Disorder
Description Here')
INSERT INTO Disability Values(019,'Quadriplegia ',' Disorder Description
Here')

INSERT INTO Membership Values(123, 'pass1','john1')
INSERT INTO Membership Values(245, 'pass2','sammy2')
INSERT INTO Membership Values(422, 'pass3','piri3')

INSERT INTO Child Values(001,'Child has tendancy to damage toys',
'English','Male', 'No Iwi', '10-09-2012','Smith', 'John',123)
INSERT INTO Child Values(002,'This child shares toys with siblings',
'Mandarin', 'Male', 'No Iwi', '11-07-2014','Mark', 'Lawrence',245)
INSERT INTO Child Values(003,'Does not like Maths toys', 'Maori',
'Female','Te Aupōuri', '06-01-2013','Wiri', 'Jo',422)

INSERT INTO Toy Values(4342, 'Buzz Lightyear', 'Toy Story Figure', '12-09-
2010',null, 32.90,25.90,'front room', 13145635,'Helmet, Lasers',
'available', 'toy')
INSERT INTO Toy Values(2345, 'Lightning McQueen', 'Car', '11-12-2011',
null, 17.90,10.90,'front room', 32423452,'Racing Wheels', 'available',
'toy')
INSERT INTO Toy Values(1235, '1001 Puzzle Pieces', 'Puzzle game', '09-03-
2016', null, 40.90,29.90,'backroom', 65389768,'1001 pieces of puzzle',
'maintainance', 'book/puzzle')
INSERT INTO Toy Values(2456, 'Hairy Maclary', 'Book', '01-04-2009',
null,22.90,20.90,'backroom', 23425233,'302 pages', 'available',
'book/puzzle')
INSERT INTO Toy Values(8976, 'Bob the Builder Kit', 'Set of Bob the Builder
tools', '11-04-2017',null, 15.90,10.90,'front room', 74563523,'Plastic
Hammer, Wrench', 'available', 'toy')

Select* from Membership
Select * from Disability
Select * from Child
Select * from Category
Select * from Toy

Drop Table Membership
Drop Table Has
Drop Table IsIn
Drop Table Lending
Drop Table Toy
Drop Table Category
Drop Table Disability
Drop Table Child
```

Representative DB tables

Toy Categories:

categoryID	title	categoryDescription
1	Active Play	Toy Description Here
2	Animals	Toy Description Here
3	Asking Questions	Toy Description Here
4	Babies	Toy Description Here
5	Balance	Toy Description Here
6	Books	Toy Description Here
7	Cars	Toy Description Here
8	CD music	Toy Description Here
9	CD story	Toy Description Here
10	CDs	Toy Description Here
11	Colour	Toy Description Here
12	Communcation	Toy Description Here

Disability Categories:

disabilityID	title	descriptionInjury
1	ADHD	Attention Deficit Hyperactivity Disorder
2	ASD	Autism spectrum disorder
3	Asthma	Disorder Description Here
4	Blind	Disorder Description Here
5	Cerebral palsy	Disorder Description Here
6	Cystic fibrosis	Disorder Description Here
7	Deaf	Disorder Description Here
8	Developmental delay	Disorder Description Here
9	Downs syndrome	Disorder Description Here
10	Dyslexia	Disorder Description Here

Member Information:

membershipID	memberPassword	loginName
123	pass1	john1
245	pass2	sammy2
422	pass3	piri3

Child Information:

childID	note	languageSpoken	gender	iwi	DoB	lname	fname	membershipID
1	Child has tendancy to damage toys	English	Male	No Iwi	10-09-2012	Smith	John	123
2	This child shares toys with siblings	Mandarin	Male	No Iwi	11-07-2014	Mark	Lawrence	245
3	Does not like Maths toys	Maori	Te Aupouri	Female	06-01-2013	Wiri	Jo	422

The above table shows an example of 3 children in the database.

Toy Information:

toyID	toyName	toyDescription	purchasedOn	replaceCost	origCost	libraryLocation	legacyID	parts	toyStatus	toyType
1235	1001 Puzzle Pieces	Puzzle game	09-03-2016	40.9	29.9	backroom	65389768	1001 pieces of puzzle	maintainance	book/puzzle
2345	Lightning McQueen	Car	11-12-2011	17.9	10.9	front room	32423452	Racing Wheels	available	toy
2456	Hairy Maclary	Book	01-04-2009	22.9	20.9	backroom	23425233	302 pages	available	book/puzzle
4342	Buzz Lightyear	Toy Story Figure	12-09-2010	32.9	26.9	front room	13145635	Helmet, Lasers	available	toy
8976	Bob the Builder Kit	Set of Bob the Builder tools	11-04-2017	15.9	10.9	front room	74563523	Plastic Hammer, Wrench	available	toy

The above table shows an example of 5 different toys in the database.

Login Code

SQL.cs

```
namespace Deliverable2_Login

{
    using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Text;
    using System.Threading.Tasks;
    using System.Data.SqlClient;
    using System.Windows.Forms;

        //Name:Andre Ofsoski
        //Student ID: 1314669
        class SQL
        {
            //generates the connection to the database
            public static SqlConnection con = new SqlConnection(@"Data
Source=cairo;Database=ajo20_Deliverable2;Integrated Security=True");
            public static SqlCommand cmd = new SqlCommand();
            public static SqlDataReader read;

            /// <summary>
            /// This excecutres the query, used mainly for
            /// insert/delete/update statements etc. where we don't need
            /// to read from what we are doing.
            /// </summary>
            /// <param name="query"></param>
            public static void executeQuery(string query)
            {
                //try catch to catch any unforseen errors gracefully
                try
                {
                    con.Close();
                    cmd.Connection = con;
                    con.Open();
                    cmd.CommandText = query;
                    cmd.ExecuteNonQuery();
                }
                catch (Exception ex)
                {
                         //error message
                        MessageBox.Show("Login attempt unsuccessful! Please check details");
                        return;                }
            }

            /// <summary>
            /// Generates an SQL query based on the input
            /// </summary>
            /// <param name="query"></param>
            public static void selectQuery(string query)
            {
                try
                {
                    con.Close();
                    cmd.Connection = con;
                    con.Open();
                    cmd.CommandText = query;
                    read = cmd.ExecuteReader();
                }
                catch (Exception ex)
                {
                         //error message
                        MessageBox.Show("Login attempt unsuccessful! Please check details");
                        return;                }
            }
```

```
        public static void editChildInfo(String childQuery, ListBox listBox)
        {
            bool clear = true;

            ////The     SQL     select  query,  using   above   string
            SQL.selectQuery(childQuery);

            //clear the listbox previous data
            listBox.Items.Clear();

            try
            {
                //Check  that    data    has     been    returned
                //Then   loop    through each    row,    printingthe     data    to      the     list
        box
                //Check  that    there   is      something       to      write   into    list    box
                if (SQL.read.HasRows)
                {
                    //listBox.Items.Add("Results for:    " + childID);
                    //loop      through each    table   row     from    the     database
                    while (SQL.read.Read())
                    {
                        //get    the     data    values  and     store   them    in      variables
                        string childID = SQL.read[0].ToString();
                        string childNote = SQL.read[1].ToString();
                        string childLanguage = SQL.read[2].ToString();
                        string childGender = SQL.read[3].ToString();
                        string childIwi = SQL.read[4].ToString();
                        string childDoB = SQL.read[5].ToString();
                        string childFName = SQL.read[6].ToString();
                        string childLName = SQL.read[7].ToString();
                        string childMemberId = SQL.read[8].ToString();

                        //display       each    of      the     rows    in      a       nice    way
                        listBox.Items.Add("child ID: " + childID + "   Notes: " + childNote + "
Language: " + childLanguage +
                            "   Gender: " + childGender + "   Iwi: " + childIwi + "   DOB: " +
childDoB + "   First Name: " + childFName
                            + "   Last Name: " + childLName + "   MemberID: " + childMemberId);
                    }
                }
                else //where    it      doesnt  have    any     successful      searches
                {
                    listBox.Items.Add("No       User    Sessionsfound.");
                }
            }
            catch
            {
                //If an error happens here, it means error in locating data
                MessageBox.Show("Error in querying database.  Please check that the database is
connected.");
            }
        }
    }

}
```

Form.cs (Login Page)

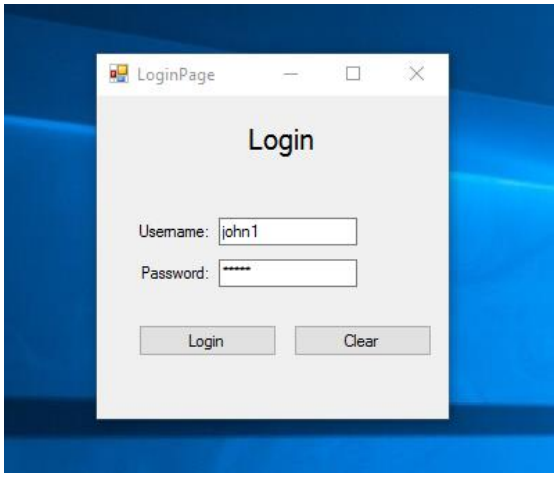
```
namespace Deliverable2_Login
{
    public partial class LoginPage : Form
    {
        public LoginPage()
        {
            InitializeComponent();
        //This line of code allows us to obscure the password visually and limit the max chars in
        textbox
            textBoxPassword.PasswordChar = '*';     //password character to hide password characters
            textBoxPassword.MaxLength = 20;         //max textbox character count
        }

        private void label3_Click(object sender, EventArgs e)
        {

        }

        private void buttonLogin_Click(object sender, EventArgs e)
        {
            //Variables to be used: 1x bool, 4x string
            bool loggedIn = false;
            string username = "", password = "";

            //check if boxes are empty, the Trim removes white space in text from either side
            if ("".Equals(textBoxUsername.Text.Trim()) || "".Equals(textBoxPassword.Text.Trim()))
            {
                MessageBox.Show("Please make sure you enter a Username and Password");
                return;
            }

            //GET the username and password from the text boxes, is good to put them in a try catch
            try
            {
                username = textBoxUsername.Text.Trim();
                password = textBoxPassword.Text.Trim();
            }
            catch
            {
                //Error message, more useful when you are storing numbers etc. into the database.
                MessageBox.Show("Username or Password given is in an incorrect format.");
                return;
            }

            //(2) SELECT statement getting all data from members, i.e. SELECT * FROM Members
            SQL.selectQuery("SELECT      *       from    Membership");

            //(3) IF it returns some data, THEN check each username and password combination, ELSE
            There are no registered users
            if (SQL.read.HasRows)
            {
                //cycle through all     users   checkingif      the     usernameexists  and     if
                the     passwordmatches
                while (SQL.read.Read())
                {
                    //SQL.read[i]:      i=0     is      UserName...     i=4     is      Password
                    if (username.Equals(SQL.read[2].ToString()) &&
password.Equals(SQL.read[1].ToString()))
                    {
                        //Username      and     Passwordcorrect,get     fname,  lname   to
                        display
                        loggedIn = true;
                        break; //stops   the     while   loop    since   they    have    logged  in
                    }
                }
            }
            else
            {
                //Error message to      show    that    no      users   have    been    registered
                MessageBox.Show("No     users   have    been    registered with these details.");
                return;
            }
```

```
            //if logged in display a success message
            if (loggedIn)
            {
                //message stating we logged in good
                MessageBox.Show("Successfully logged in as: " + username);
                initialiseTextBoxes();
                this.Hide();
                ChildInfo childPage = new ChildInfo();
                childPage.ShowDialog();
                this.Close();
            }
            else
            {
                //message stating we couldn't log in
                MessageBox.Show("Login attempt unsuccessful! Please check details");
                textBoxUsername.Focus();
                return;
            }
        }

        public void initialiseTextBoxes()
        {
            //goes through and clears all of the textboxes
            foreach (Control c in this.Controls)
            {
                if (c is TextBox)
                {
                    (c as TextBox).Clear();
                }
            }
            //makes next place user types the text box
            textBoxUsername.Focus();
        }

        private void buttonClear_Click(object sender, EventArgs e)
        {
            textBoxPassword.Clear();
            textBoxUsername.Clear();
            textBoxUsername.Focus();
        }


        private void buttonBrowse_Click(object sender, EventArgs e)
        {

            Hide();
            ChildInfo browse = new ChildInfo();
            browse.ShowDialog();
            Close();
        }
    }
```

ChildInfo.cs

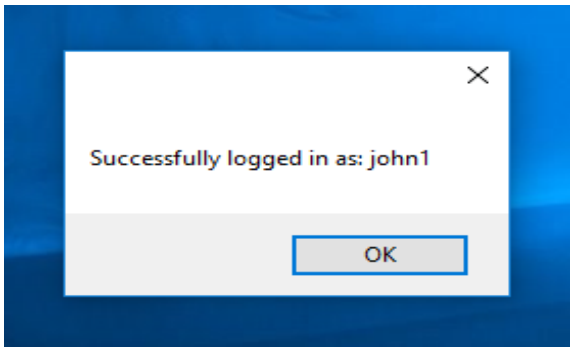
```
namespace Deliverable2_Login
{
    public partial class ChildInfo : Form
    {
        public ChildInfo()
        {
            InitializeComponent();
            string childQueryInfo= "Select * from Child";
            SQL.editChildInfo(childQueryInfo, listBoxChildInfo);
        }


        private void ChildInfo_Load(object sender, EventArgs e)
        {

        }
    }
}
```

Walkthrough



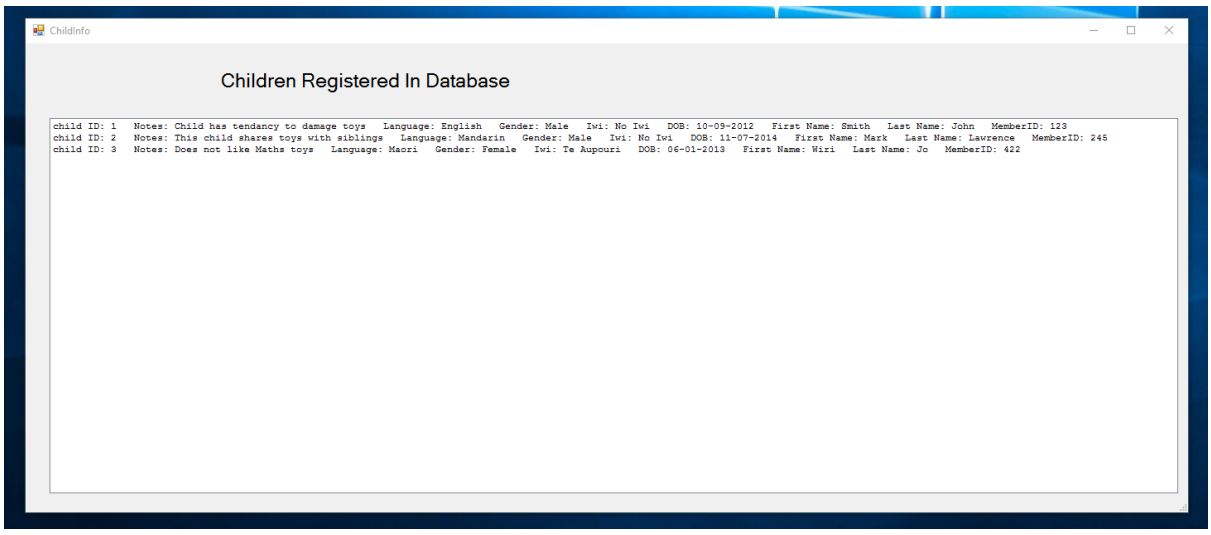
1. To login to the database and view child data you need to type your username into the “Username” textbox and then your password into the “Password” textbox, then press the login button.

   If you make any mistakes in this process press the clear button to restart.



2. If you have successfully entered your appropriate details you will receive a similar message with your username instead of “john1”.

   Now press OK to continue.



3. After you have pressed OK, the above screen should appear displaying all the relevant child information in the database.