ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ

Факултет Компютърни системи и технологии

Катедра „Компютърни системи“

**ДИПЛОМНА РАБОТА**

за придобиване на образователно-квалификационна степен   
„**бакалавър/магистър**“

на тема

**„Data center“**

|  |  |
| --- | --- |
| *Дипломант:* | *Научен ръководител:* |
| Ali Fares Chakroun  Фак. номер: 273217006  Група: 1  Специалност: Computer science and engineering | титла и имена |

София, 2021 г.

# Table of Content

[1. Why use database 3](#_Toc516145954)

[1.1. Small bussiness owners 3](#_Toc516145955)

[1.1. Purpose of the application 3](#_Toc516145955)

[2. Technologies used 4](#_Toc516145961)

[2.1.Windows form application 4](#_Toc516145962)

[2.2. Zen.barcode.core 4](#_Toc516145963)

[2.3. Aforge 4](#_Toc516145964)

[2.4. MySQL database 5](#_Toc516145965)

[2.5. Why use these technologies 5](#_Toc516145965)

[3. Design 6](#_Toc516145966)

[3.1.Purpose of the application 7](#_Toc516145962)

[3.2.What data is used and how 7](#_Toc516145962)

[3.3.Functionality 8](#_Toc516145962)

[4. Code.. 19](#_Toc516145967)

[4.1.Windows form(c#) 19](#_Toc516145962)

[4.2.MySQL table 27](#_Toc516145962)

[5. Guide 30](#_Toc516145968)

[5.1.Passcode 30](#_Toc516145962)

[5.2.Mainpage 31](#_Toc516145962)

[5.3.Add Item 32](#_Toc516145962)

[5.4.Search 34](#_Toc516145962)

[5.5.Camera 36](#_Toc516145962)

[5.6.Item information 38](#_Toc516145962)

[5.7.Delete 41](#_Toc516145962)

[5.8.Settings 44](#_Toc516145962)

[6.Conclusion 47](#_Toc516145969)

[7.Reference. 48](#_Toc516145970)

1. **Why use database.**
   1. **Small Business owners**

Bussiness owners owning a warehouse often need to spend a lot of time and man power checking their warehouses for products stored, whether it is the item itself or the quantity of the item it relies on a lot of man power and time.

It can be used to double check the paperwork to see if the system or papers have different information in them like the quantity of an item. It will show them what they are missing and how much.

Using software gives owner an easy and simple way to keep track of their products and how much was sold over an x duration.

* 1. **Purpose of the application**

Purpose of the application is to help warehouse owner keep track of their products and to update them when ever necessary.

Goals:

* To give the warehouse owners an easy way of accessing their warehouse items without the need to search for the item via papers/documents.
* To have an easier way of finding the quantity of an item(so you don’t have to recount everytime).
* To double check the database information with your documents so you know if your missing an amount of an item.
* Gives the user a list of all the changes that happened to an item.
* To make price changing easy and fast.
* To give employees a way to check the quantity of an item so that they can sell to customers who want to order a lot of.

Task:

* Allows users to add new items into the database.
* All items have a unique barcode which can be scanned using the camera to get instant full access to all information connected to the barcode.
* Gives the users a fast way of getting information of an item with a single click.
* Allows users to update information of an items via the search bar or with camera.
* Allows users to Delete Items form the database.

1. **Technologies used.**
   1. **Windows Form Application**

Is an open-source graphical (GUI) class library included as a part of Microsoft .NET, .NET Framework or Mono Framework, providing a platform to write rich client applications for desktop, laptop, and tablet PCs. While it is seen as a replacement for the earlier and more complex C++ based Microsoft Foundation Class Library, it does not offer a comparable paradigm and only acts as a platform for the user interface tier in a multi-tier solution

**2.2 Zen.Barcode.Core**

.NET Core Barcode is a Portable Class Library (PCL) available in the ConnectCode Barcode Fonts package that generates barcodes that meet the strictest requirements of the auto-id industry. It is a cross-platform Portable Class Library that supports Windows, macOS and Linux, and can be used in device, cloud, and embedded/IoT scenarios.

As a barcode font raster to the output device and are not limited to DPI (Dots per Inch) of the computer screen, a barcode generated using fonts is of the highest quality and can meet the strictest requirements required by the auto-id industry.

**2.3 Aforge:**

Is an open source C# framework designed for developers and researchers in the fields of

Computer Vision and Artificial Intelligence - image processing, neural networks, genetic algorithms, fuzzy logic, machine learning, robotics, etc.

The framework is comprised by the set of libraries and sample applications, which demonstrate their features:

AForge.Imaging - library with image processing routines and filters;

AForge.Vision - computer vision library;

AForge.Video - set of libraries for video processing;

AForge.Neuro - neural networks computation library;

AForge.Genetic - evolution programming library;

AForge.Fuzzy - fuzzy computations library;

AForge.Robotics - library providing support of some robotics kits;

AForge.MachineLearning - machine learning library;

The work on the framework's improvement is in constants progress, what means that new feature and namespaces are coming constantly. To get knowledge about its progress you may track source repository's log or visit project discussion group to get the latest information about it.

The framework is provided not only with different libraries and their sources, but with many sample applications, which demonstrate the use of this framework, and with documentation help files, which are provided in HTML Help format.

**2.4 MySQL database:**

Is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses.

**2.5 Why choose these technologies?**

**Windows Form Application** is the ideal easy to use user interface(UI) framework. It provides one of the most productive ways to create desktop apps based on the visual designer provided in Visual Studio. Functionality such as drag-and-drop placement of visual controls makes it easy to build desktop apps.

**Zen.Barcode.Core** allows the creation of a barcode.

**AForge** allows the use of cameras with the help of (AI) it makes it possible to scan and read the barcodes.

**MySQL** one of the best database management systems allowing us to easly manipulate its table meat our needs.

1. **Design**
   1. **Purpose of the application**

The main goal of this application is to help owners with warehouses a way to keep trach of their products and the amount of each product whether it is to add more of the same product or to remove some from it.

It can also be used in stores, by adding a barcode label to each item it allows employees to have instant access to the items information, giving them the ability to check the items quantity before selling high number of pieces to a customer if there isn’t enough in the warehouse.

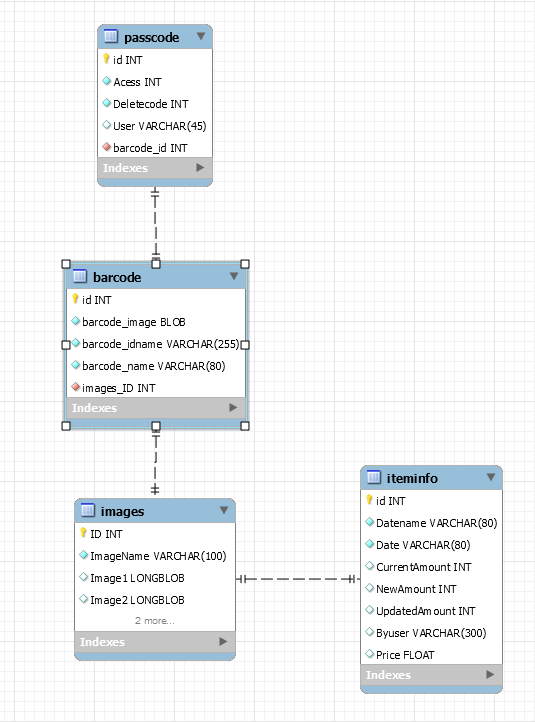
* 1. **What data is used and how.**

When the user wants to insert data into the database or update the database the data/values he inputs gets saved in its own variable, then these values are all inserted into a query that is then executed and send to the database.

When the user inputs data it gets processed behind the scene (back end) of the application if no errors occur it gets redirected into a query that executes the commands to ADD/UPDATE/DELETE from the database.

It is a constant communication between the application and the database.

The structure of the database is as follows:



Our first table “passcode” checks for registered users in the database when it find it will then allow access to the other tables of the system.

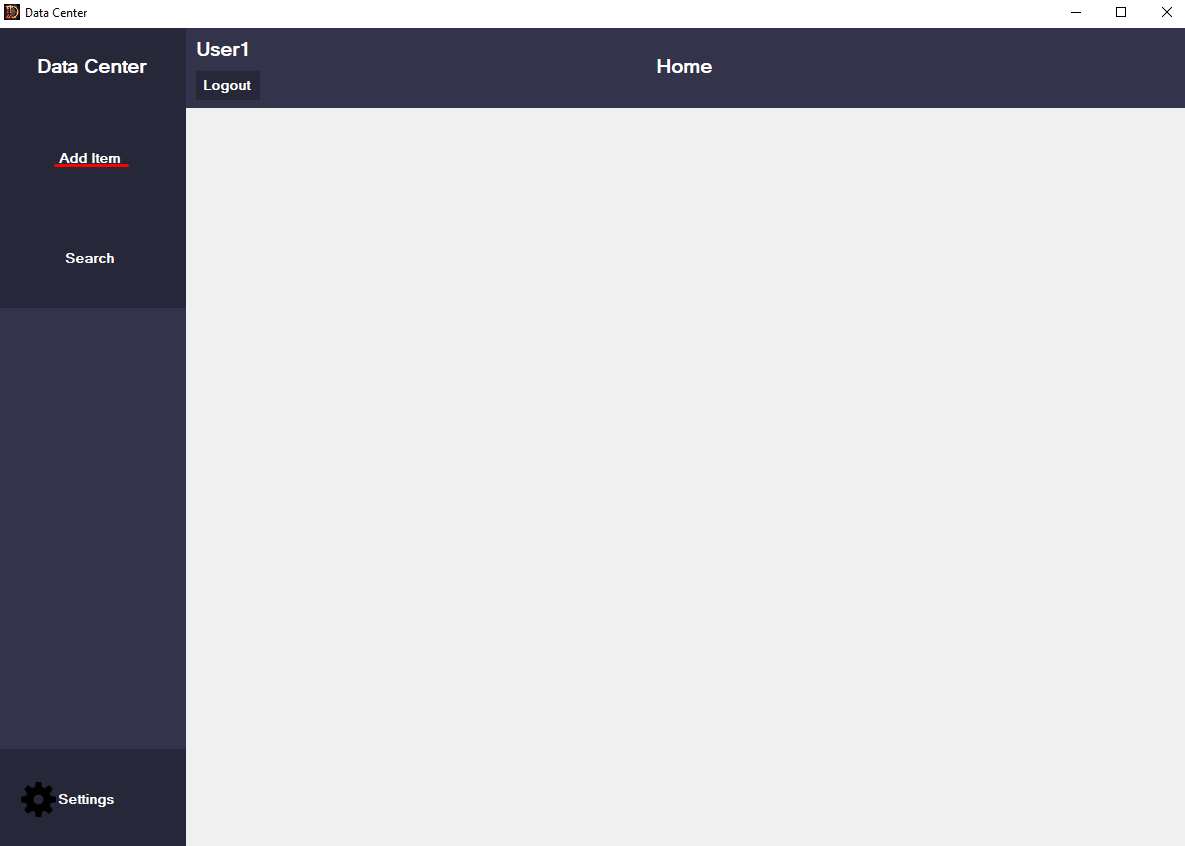
When the application send a request for an items name it check the follow:

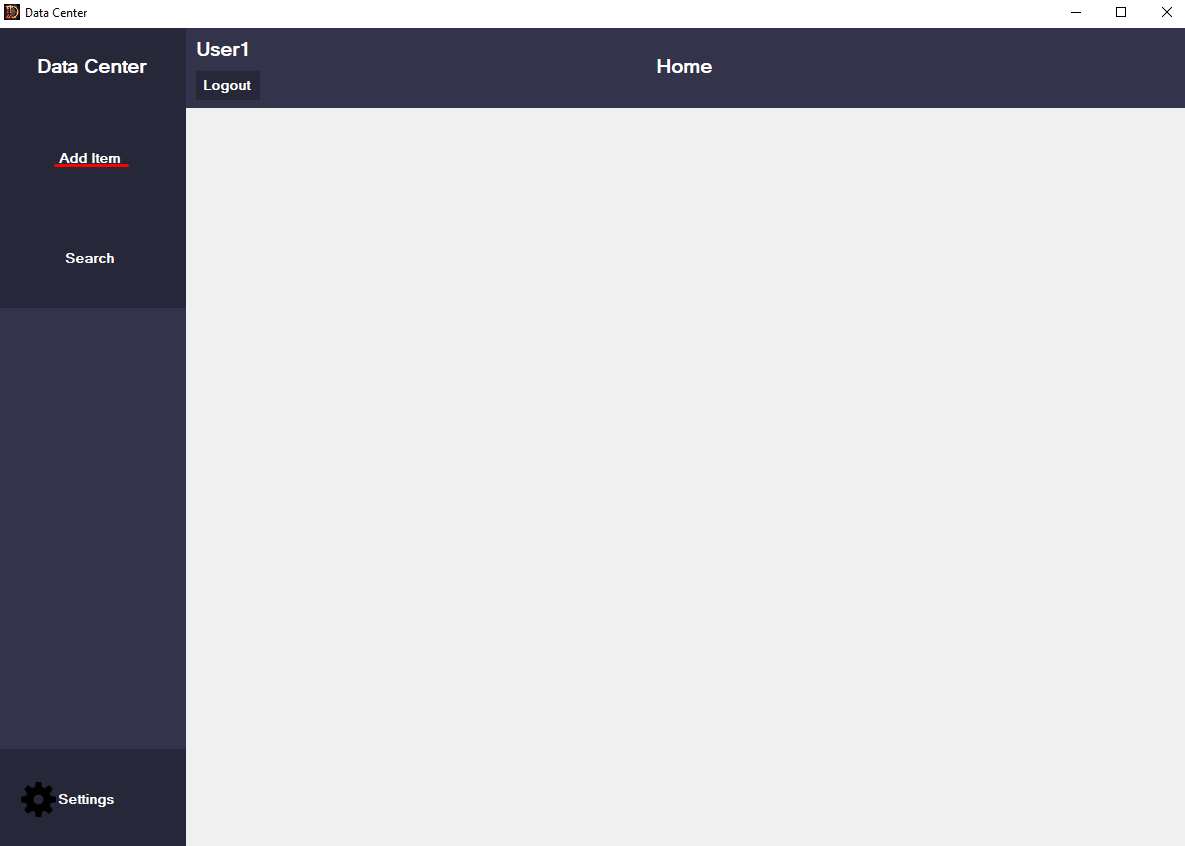
* Barcode\_name.
* ImageName.
* Datename

These 3 values are what is used to find and send the information to the user to see.

* 1. **Functionalities**

The Application is designed for businesses owning a warehouse and need an efficient way to keep track of their products.

To add items to the database all the user need to do is click the “Add item” button which then will take him to a page where he will be asked to fill the missing information.



* A unique name that will be inserted to the database(no same names can be added into the database).
* (optional)to put how much of the item you have(this can be updated later on) by default its going to be 0.
* Price of the item(can be updated later on)default is 0;
* (optional)you can put images of the item for easier and faster identification when searching for the item manually(Max 3 images).

After inserting all information you can choose to automatically save it or to manually save it.

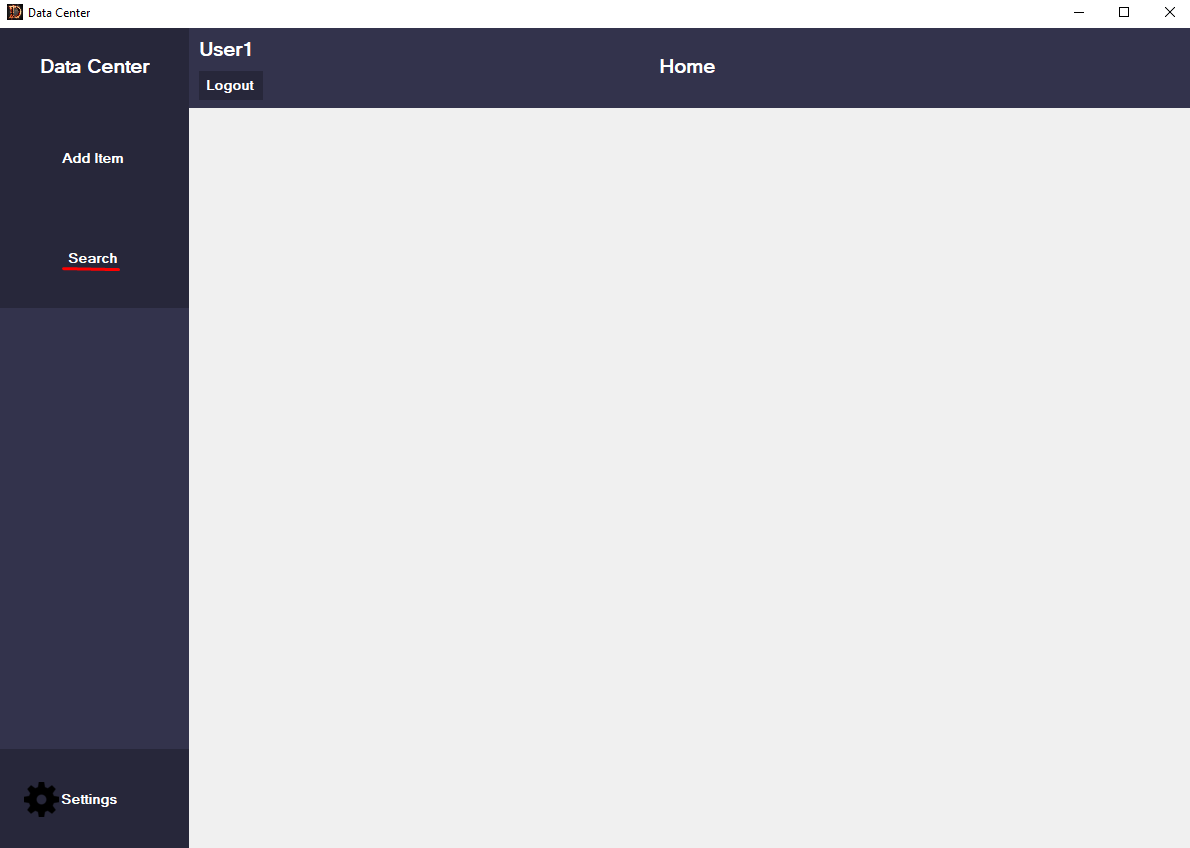
Pressing the “Save” button will automatically save while pressing the “Save at” will allow you to manually choose where to save the uniquely generated barcode that has all the items information saved at.

After a successful save the user will get a barcode that can be scanned for information of the item.

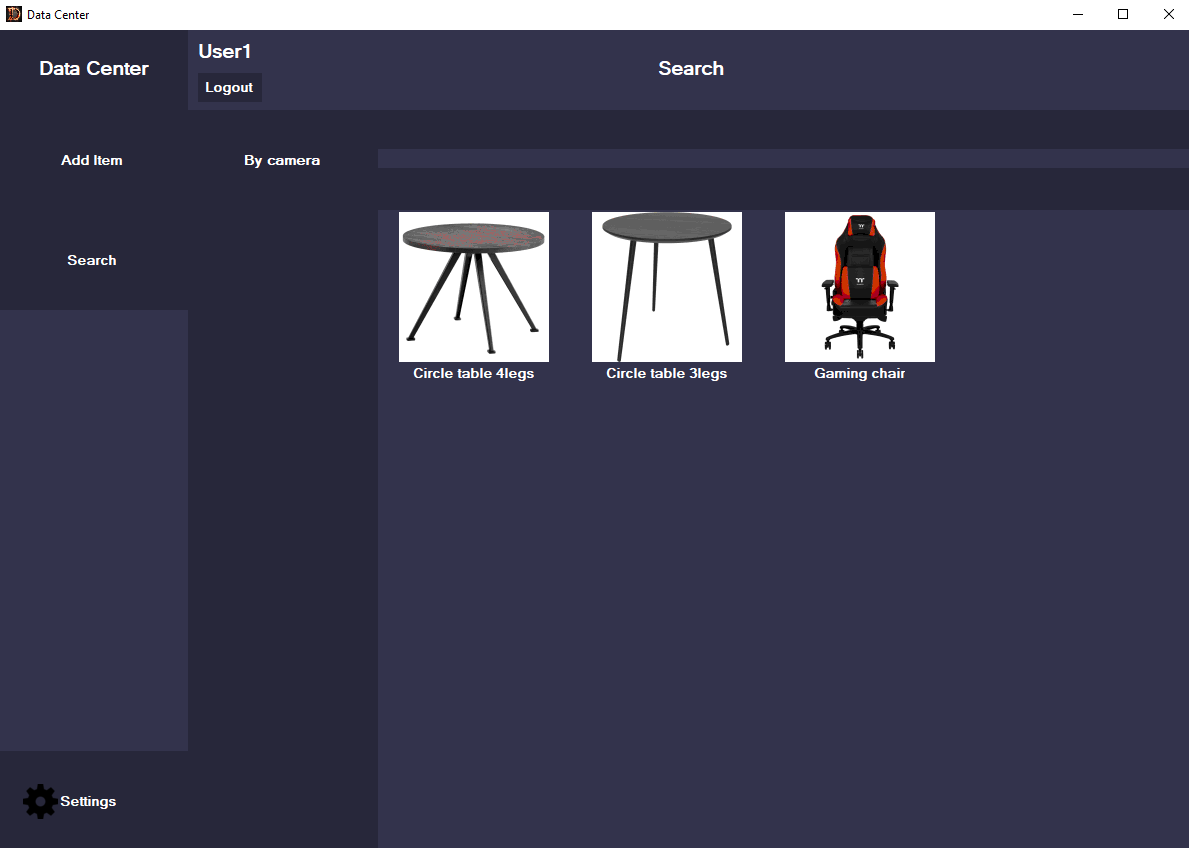


Searching for existing items in the database is as easy as adding items.

First you press on the “Search” button.

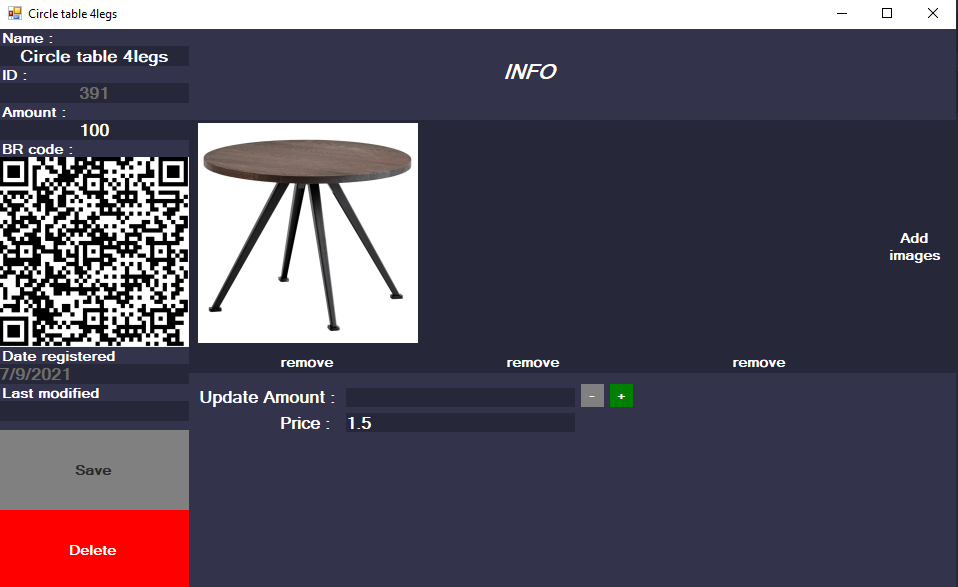


You will then be taken to a page with a search bar and a list of all the items in the database.



You can choose to search for item by writing the items name in the search bar or by browsing the list that contains all the items in the database.

When you find the item your looking for you need to do is press on the item to get a pop up with all information related to the item.

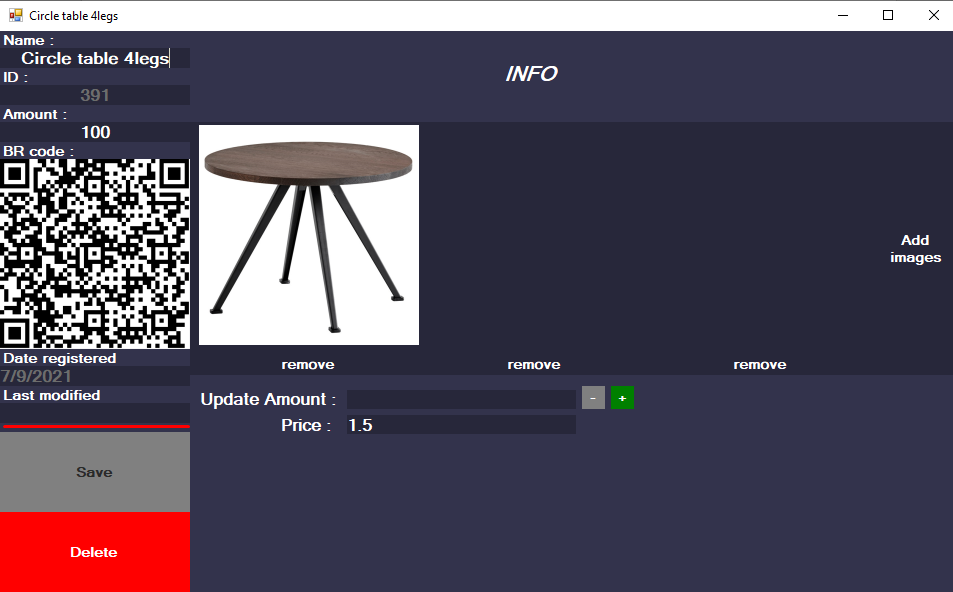


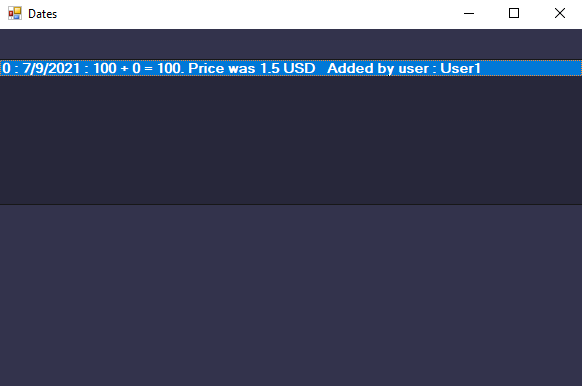
After the items information loads in you can :

* Add/subtract from the items quantity.
* Change the price of the item.
* Change the name of the item.
* Update/remove the images if the item.

It also shows you the date the item was first created on and the last date it was updated in.

You could also check a list of all the changes occurred to the item by clicking the bar under “Last modified”





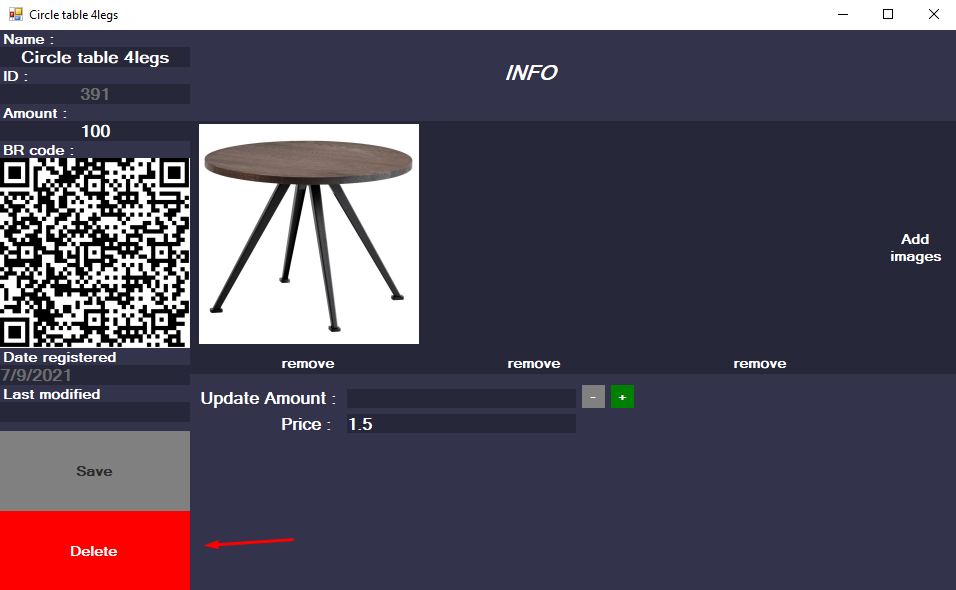
On the Dates popup you will see:

* Date in which the item was change in.
* The original Amount +/- the new amount
* Price at the time of the change
* By which user the item was updated with.

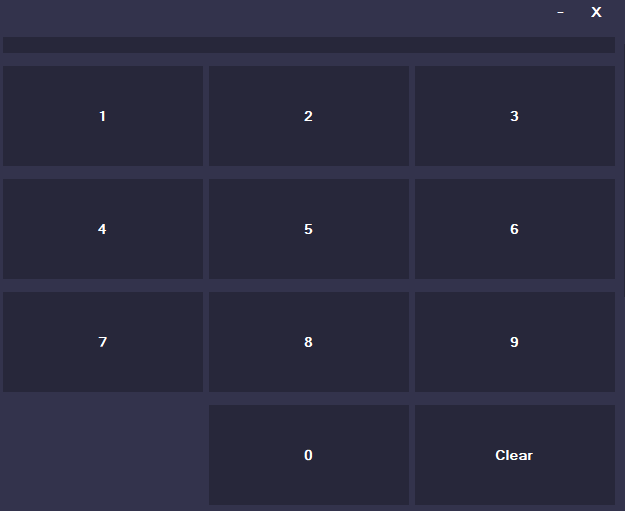
To delete an item the user must have a delete passcode

Not all account have the delete passcode meaning only authorized users can delete from the database.

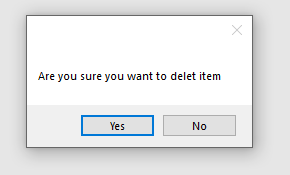
To delete item all they need is to press the delete button.



After pressing the delete button the user will get a pop up asking for the delete code



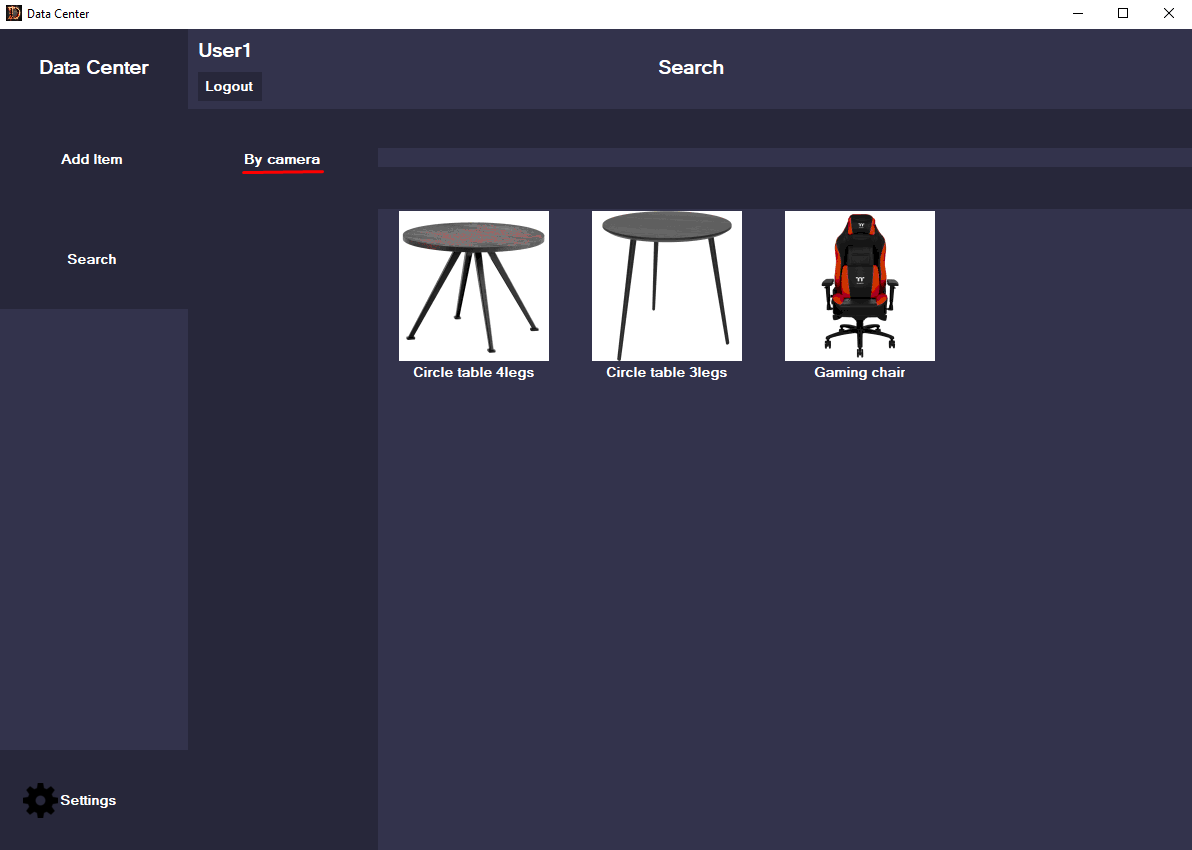
On a successfully inserting the delete code you will get a final verification

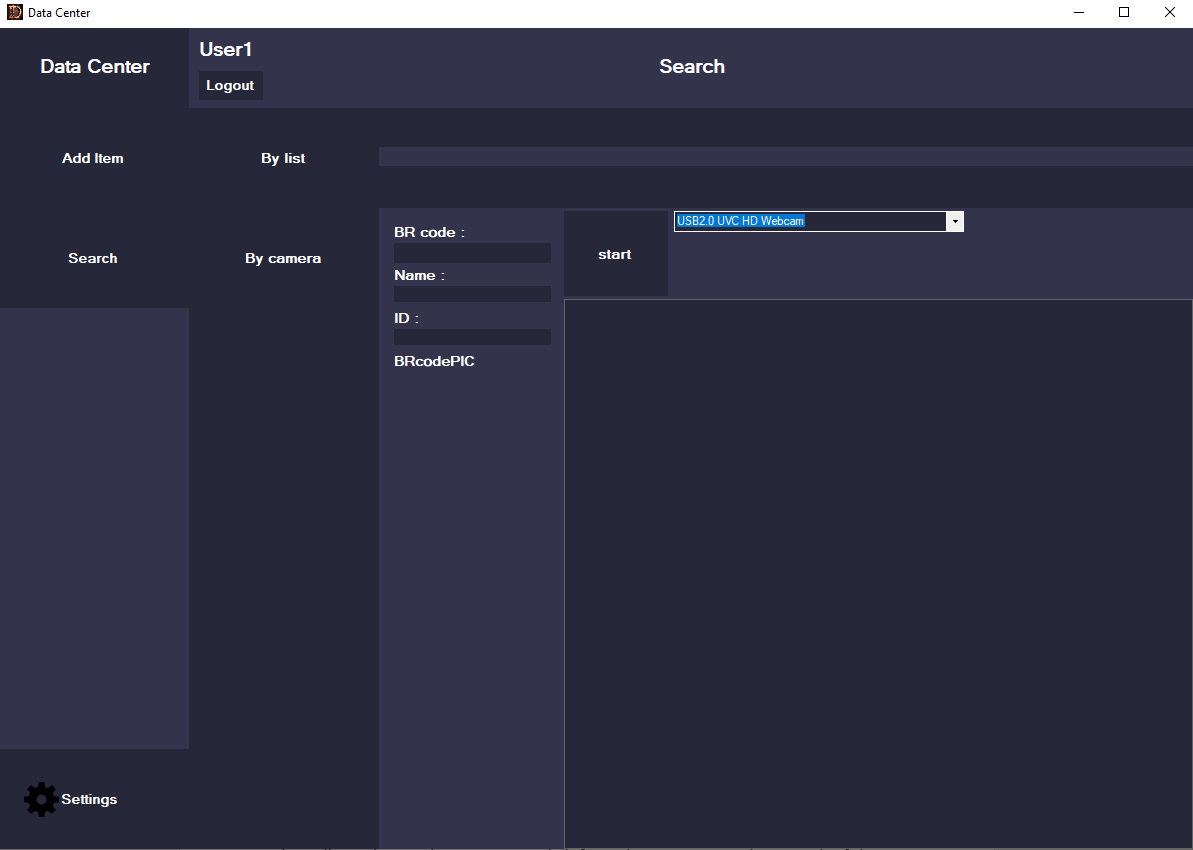


Pressing yes will delete the item form database.

To get all information related to an item you could also scan the barcode given to the user when creating an item.

To scan it simply press the “By camera” button in the ”Search” page.





Here you can see:

* The barcode name
* The Name of the item
* ID of the item
* Picture of the barcode
* A list of all cameras connected to your computer.

By giving employees a fast way of getting all information related to an item they can check the quantity of the item they have in their warehouse without the need for them to search for papers containing this information reducing the time needed.

1. **Code**
   1. **Windows form (c#)**

The main code connecting the Application to the database is through a class called “DBconn” inside the protected class “DBconnection” is what connects the application to the server.

protected string DBconnection()

{

string connString = "server=localhost;user id=root;password=root123;persistsecurityinfo=True;database=barcode";

return connString;

}

For communication between the database and the application the information gets transferred through switch cases.

protected string DBstrings(int x) {

string Dbstr = " ";

switch (x)

{

case 1:

Dbstr = "insert into barcode(id, barcode\_image, barcode\_idname, barcode\_name) values(@idnum, @image, @idname, @name); insert into images(ID, ImageName, Image1, Image2,Image3) values(@imageID, @ImageIDnum, @Image1num, @Image2num,@Image3num); insert into iteminfo(id, Datename, Date,CurrentAmount,NewAmount,UpdatedAmount,Byuser,Price) values(@idnum, @name, @Date1,@Amount1,@Amount2,@Amount3,@user,@price);";

break;

default:

MessageBox.Show("Error no MYSQL command found");

break;

}

return Dbstr;

}

This query is whats communicates with the database to add items into the database.

To extract information form the database we use this query.

case 3:

Dbstr = "select \* FROM barcode INNER JOIN barcode.images ON barcode\_name = ImageName INNER JOIN iteminfo ON barcode\_name = Datename where barcode\_name = @name ORDER BY iteminfo.id ASC ; ";

break;

To update information that exist in the database we use this query.

case 4:

Dbstr = "Update barcode,images,iteminfo SET barcode\_name = @newname,ImageName =@newname,Image1 = @image1,Image2 = @image2, Image3 = @image3 ,Datename =@newname where barcode\_name =@name and ImageName = @name and Datename=@name;insert into iteminfo(Datename,Date,CurrentAmount,NewAmount,UpdatedAmount,Byuser,Price) values(@newname,@date,@Amount1,@Amount2,@Amount3,@user,@price);";

break;

This protected class can only be access with a number(intiger) with the help another class.

public MySqlCommand DBcmd(int x, MySqlConnection conn)

{

MySqlCommand cm = new MySqlCommand(DBstrings(x), conn);

return cm;

}

Creating a barcode:

To create barcode we first create the string of the barcode to create it we use a loop that randomly choose 1 character of the following characters in the string

For a total of 10;

public static string RandomString()

{

Const string chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

return new string(Enumerable.Repeat(chars, 10)

.Select(s => s[random.Next(s.Length)]).ToArray());

}

After creating our barcode string we then generate a barcode image

Using the Zen.barcode.Core library

Zen.Barcode.CodeQrBarcodeDraw QRcode = Zen.Barcode.BarcodeDrawFactory.CodeQr;

BRimg = QRcode.Draw(barcode, 50);

For adding images into the application we use OpenFileDialog

OpenFileDialog o = new OpenFileDialog();

o.Multiselect = true;

o.DefaultExt = ".Jpg";// Default file extension

o.Filter = "Image (.jpg;png;bmp)|\*.jpg;\*.png;\*.BMP"; // Filter files by extension

o.InitialDirectory = st.saveDir();

if (o.ShowDialog() == DialogResult.OK)

{

string filename = o.FileName;

p.SizeMode = PictureBoxSizeMode.StretchImage;

p.Image = Bitmap.FromFile(filename);

}

Here we are allowed to choose JPG,PNG,BMP image formats to be displayed in the application and to be saved in the database. To save it into the database we first convert the image into bytes and save it in the database as a blob of bytes

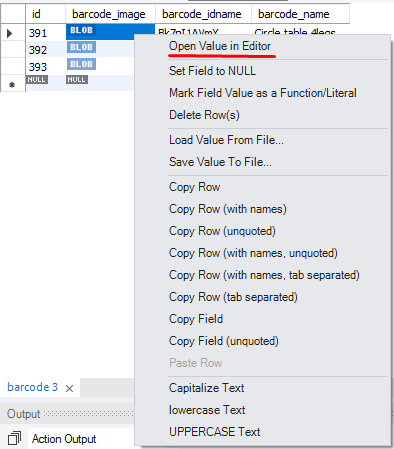
Image img1 = BRimg;

byte[] arr1;

ImageConverter converter = new ImageConverter();

arr1 = (byte[])converter.ConvertTo(img1, typeof(byte[]));

inside the database we can check how long the data is my checking the values of the blob.





To insert values into the database we use “MySqlCommand” to prepare our values and then we execute them all at once using “cm.ExecuteNonQuery”.

MySqlCommand cm = db.DBcmd(1, con);

// Create and set the parameters values

cm.Parameters.Add("@idnum", MySqlDbType.Int32).Value = null;

cm.Parameters.Add("@image", MySqlDbType.Blob).Value = arr1;

cm.Parameters.Add("@idname",MySqlDbType.VarChar).Value= Name2;

cm.Parameters.Add("@name", MySqlDbType.VarChar).Value = Name;

int rowsAdded = cm.ExecuteNonQuery();

using MySqlCommand we are able to assign value to parameters we want to add into the database after assigning each parameter we then ask the database to execute the command and insert the information into the database.

All this is located inside a try – catch statements .

To read information form the database we use MySqlDataReader read information from the database and to set its values in the application.

MySqlDataReader myreader = cm.ExecuteReader();

We then use a while loop to go through all the row that got affected by the query and we send to the database.

while (myreader.Read())

will always loop if there is a row to read.

Some of the methods to extract information from the rows are :

* myreader.GetInt32
* myreader.GetString;
* myreader.GetBytes;
* myreader.GetChars;
* myreader.GetBoolean;

etc…

Adding items to be display as a box of image and a text for easier identification

listView1.LargeImageList = imgs;

foreach (string item in imagnames)

{

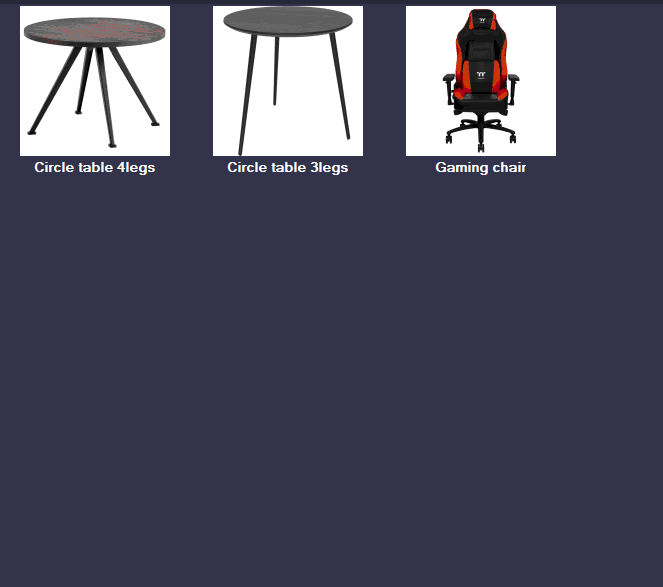
listView1.Items.Add(item, y);

y++;

}

We insert each image into the array and they loop it to add it inside the listview with its text.

Allows us to create this.



To scan barcode with a camera we use “VideoCaptureDevice” library and

“FilterInfoCollection” library to find all cameras connected to the computer.

filterInfoCollection = new FilterInfoCollection(FilterCategory.VideoInputDevice);

foreach (FilterInfo Device in filterInfoCollection)

try

{

comboBox1.Items.Add(Device.Name);

}

catch {

MessageBox.Show("no camera found" );

}

comboBox1.SelectedIndex = 0;

videoCaptureDevice = new VideoCaptureDevice();

after getting all devise we can choose which device we want to stream from.

Afterwords we point the can camera on the barcode we want to scan.

On a successful scan the camera will stop and you will get all information related to the barcode.

BarcodeReader br = new BarcodeReader();

Result res = br.Decode((Bitmap)pictureBox1.Image);

if (res != null) {

textBox1.Text = res.ToString();

timer1.Stop();

if (videoCaptureDevice.IsRunning == true)

first we check if we detect a barcode image when we do we turn the camera off and scan the barcode.

videoCaptureDevice.Stop();

string brcode = textBox1.Text;

MySqlConnection con = db.DBconnect();

con.Open();

MySqlCommand cm = db.DBcmd(5, con);

cm.Parameters.Add("@barcode", MySqlDbType.VarChar).Value = brcode;

MySqlDataReader myreader = cm.ExecuteReader();

We get the string of the barcode that was scanned and send it to the database if the database find information about the barcode it will display it , but if it doesn’t it will give you a pop up warning that it found nothing.

MySqlDataReader myreader = cm.ExecuteReader();

while (myreader.Read())

{

BRName.Text = myreader.GetString("barcode\_name");

ID.Text = myreader.GetString("id");

ItemID = ID.ToString();

Byte[] img = (Byte[])myreader["barcode\_image"];

MemoryStream ms = new MemoryStream(img);

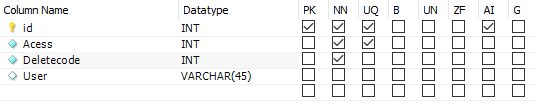
BRcodePIC.Image = Image.FromStream(ms);

}con.Close();

* 1. **MySQL tables**

We have 4 tables that communicate with the application to give the user the information they need.

We first start with the first table “passcode”



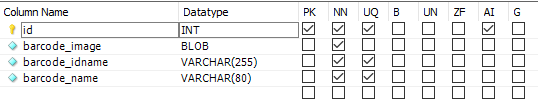
Here we have 4 values:

* Id: a number that automatically increases every time a new row is created.
* Acess: a passcode created to give access to the system.
* Deletecode: is optional giving the user the ability to delete from the database.
* User: the username of the user.

From this table the application calls for it when information when a login in request is send

Only when the passcode is correct will the user be given access.

The main table is “barcode”

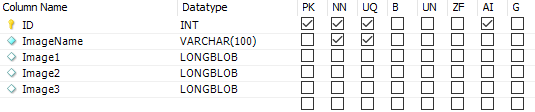


Here we have 4 values:

* Id: a number that automatically increases every time a new row is created.
* Barcode\_image: the image of the barcode that is generated by the software and saved in here.
* Barcode\_idname: the name of the barcode.
* Barcode\_name: the name given to the item by the user

“Barcode\_name” is the value that gets shares in all the tables.

Third table “Images”

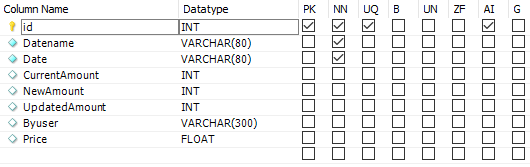


Here we have 5 values:

* Id: a number that automatically increases every time a new row is created.
* ImageName: its value is set from the “barocde” table “Barcode\_name”.
* Image1:given by the user from the application default is null.
* Image2:given by the user from the application default is null.
* Image3:given by the user from the application default is null.

Our last table “iteminfo”

Is where all the information about an item is stored.



Here we have 8 values:

* Id: a number that automatically increases every time a new row is created.
* Datename: its value is set from the “barocde” table “Barcode\_name”.
* Date:the day the item was updates or inserted, it gets it value form the software.
* CurrentAmount: the current quantity of an item set by user. Default 0
* NewAmount: the amount to be added or subtracted form the item set by user.Default is 0;
* UpdatedAmount: the total quantity of an item after calculating “CurrentAmount” and “NewAmount” values.
* Byuser: By which user the information got updated by.it gets its value from “passcode” table “User”
* Price: the price of the item given by the user. Default is 0.

MySQL allows for custom flags for each value in the database.

There are 8 flags :

* **PK** − Primary Key
* **NN** − NOT NULL
* **BIN** − Binary
* **UN** − Unsigned
* **UQ** − Unique
* **ZF** − Zero Filled
* **G** − Generate Column
* **AI** − Auto Increment

**PK**

This stands for the primary key. It can be used to make the column as a primary key.

**NN**

It is for NOT NULL. Used to enforce the column that it will not insert a NULL value.

**BIN**

This stands for Binary. This can be used to store data as a binary string.

**UN**

It is for Unsigned and can be used to store an only positive value which can be started from 0.

**UQ**

UQ is for Unique. This can be used to enforce the column to insert only unique value for a specific column.

**ZF**

ZF is for Zero Filled. Suppose, we have declared int(3) and you want to store 21, then zero filled would output the result 021.

**G**

G stands for Generated column.

**AI**

AI is for AutoIncrement.

These flags allow us to customize the each row to what we need it for

For instance we use the **Not null(NN)** flag to tell mysql that the value for this object should not be empty. Another important one is the **Unique(UN)** which always check if the value being inserted into the database is repeated if it is then the system will give us a warning.

If we want to manually insert information using just a query into these 3 table(barcode,images,iteminfo)

Then we can use this query:

insert into barcode(id, barcode\_image, barcode\_idname, barcode\_name) values(null, 1,"brstring", "Example1"); insert into images(ID, ImageName, Image1, Image2,Image3) values(null, "Example1", null, null,null);

insert into iteminfo(id, Datename, Date,CurrentAmount,NewAmount,UpdatedAmount,Byuser,Price) values(null, "Example1", "now",100,0,100,"Test1",12.99);

when executing this code we will get a new row if our values in them. To access this row we can use the INNER JOIN method to get it all.

select \* from barcode Inner JOIN barcode.images ON barcode\_name = ImageName INNER JOIN iteminfo ON barcode\_name = Datename where barcode\_name = "Example1" ;

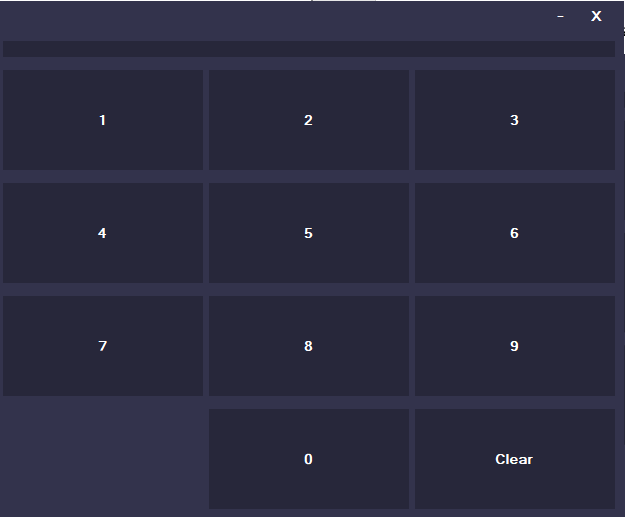
After executing this query we get



This is how the application receives and send information through queries.

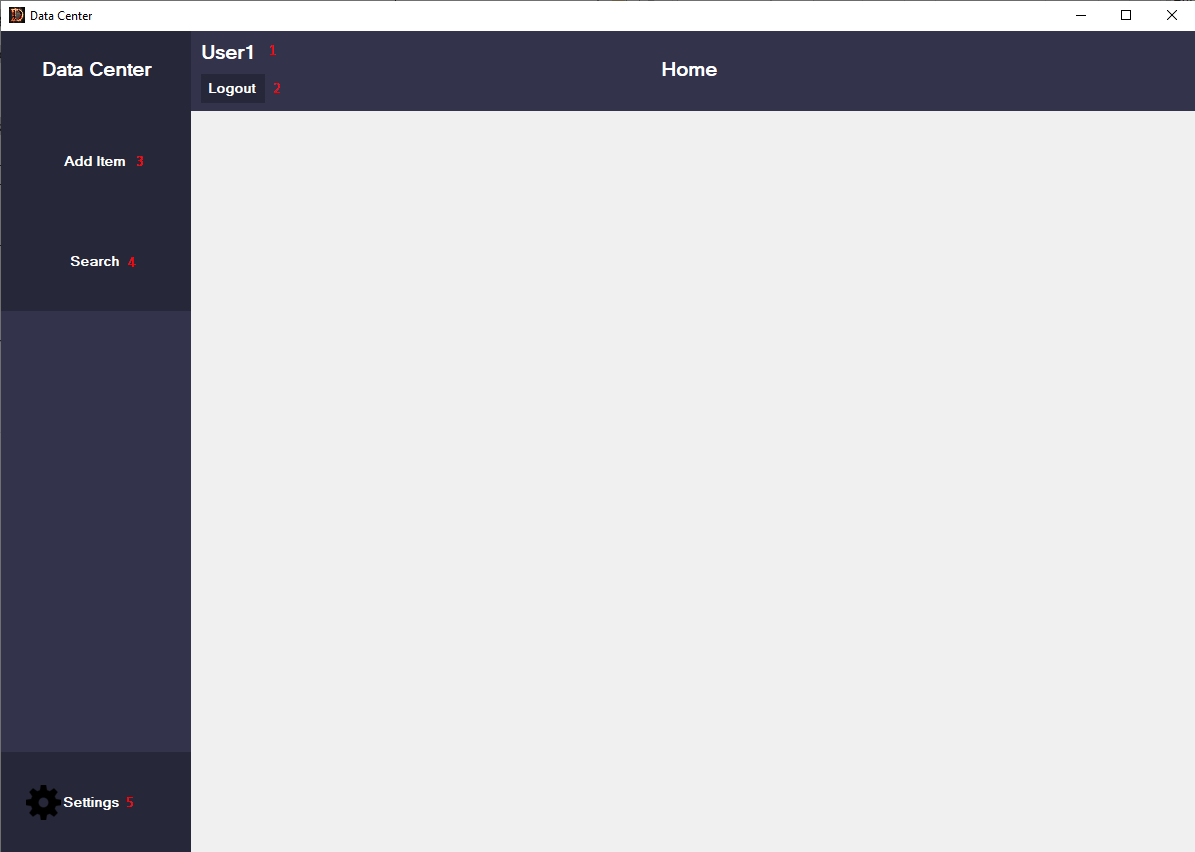
1. **Guide**
   1. **Passcode**

When you start the application it first will ask for a password(each account has it own unique password).



After a successful verification you will me send to the mainpage.

* 1. **Mainpage**



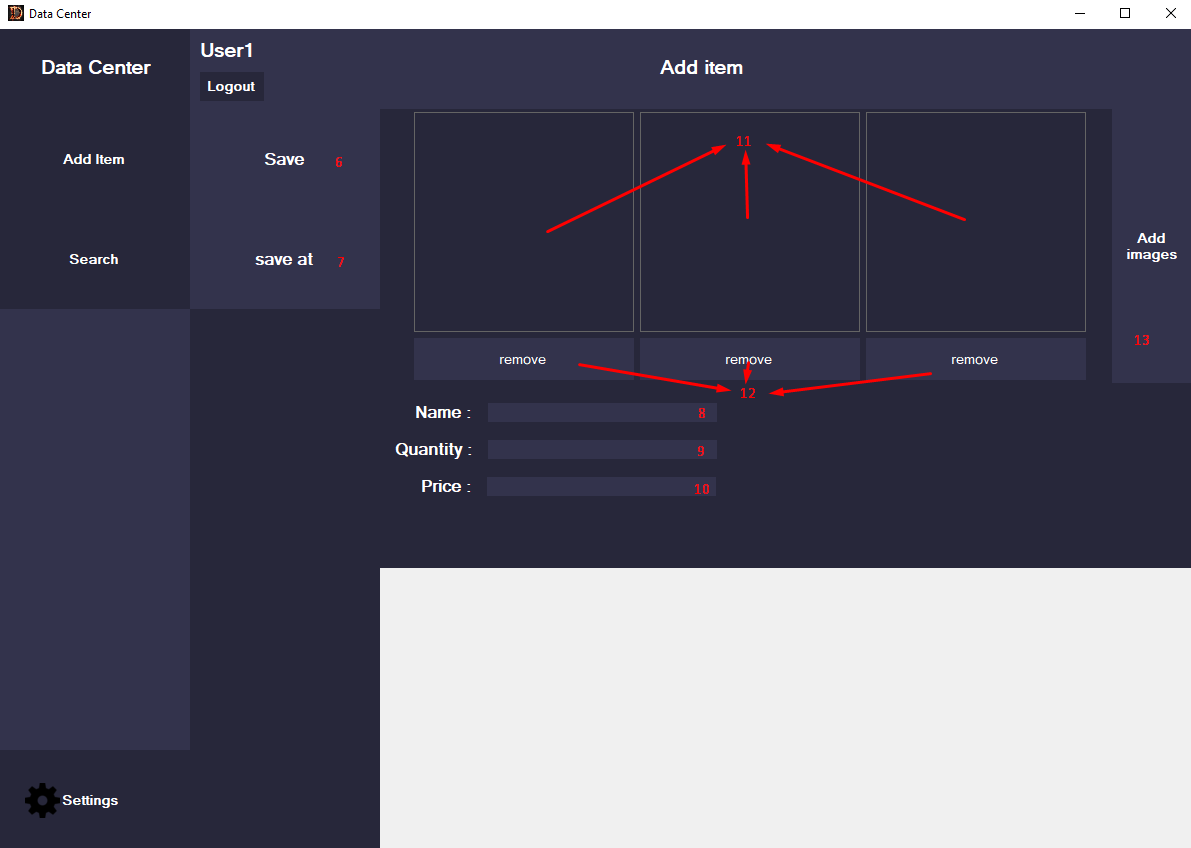
* + - 1. Accounts name.
      2. Logout from the system.
      3. Add item page.
      4. Search Page.
      5. Settings.

(2) Pressing the “Logout” button will log the user out and ask for a passcode to re-enter the program.

* 1. **Add item**

(3) If the user wants to add a new item into the system then click the “Add item”[3]

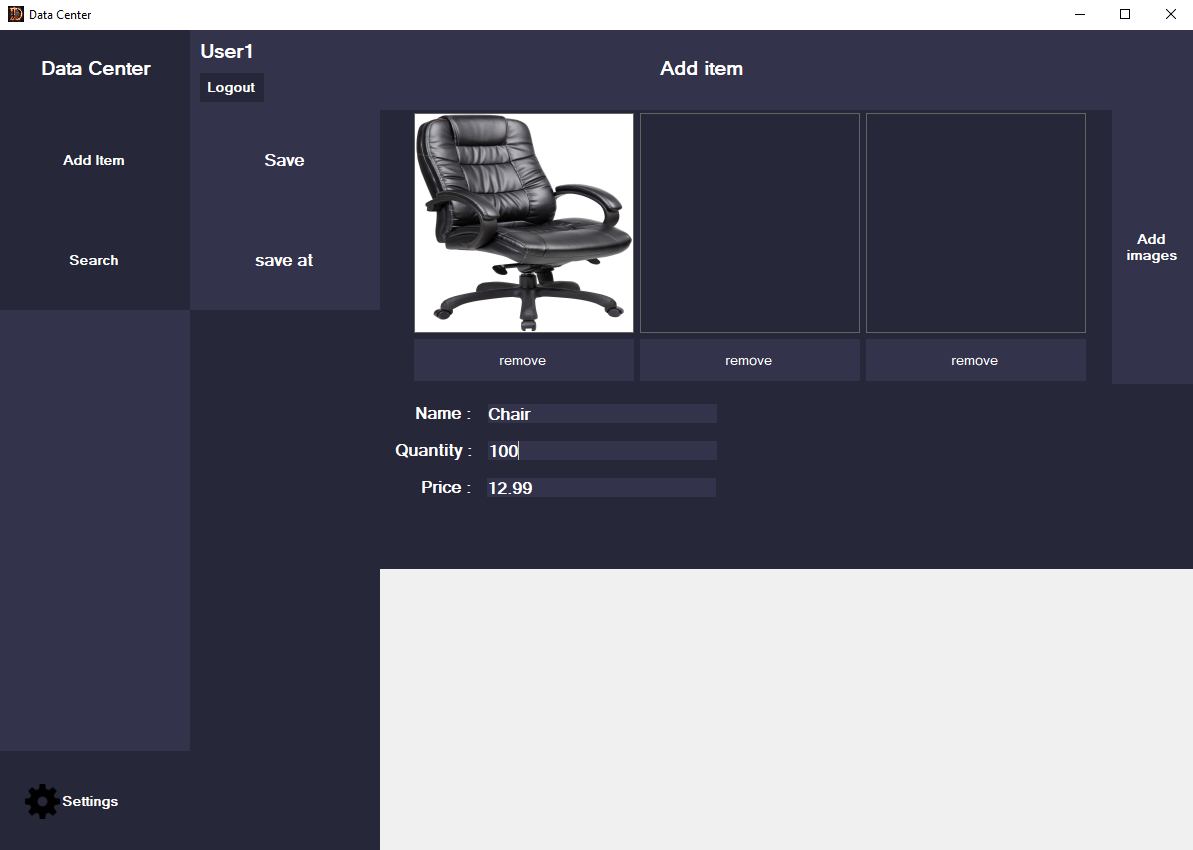
Button.



1. Saves the items and its barcode image automatically.
2. Save at a location manually.
3. Give the item its own unique name.
4. Quantity of the item.
5. Price of the item.
6. Add image to the items information.
7. Remove image from the items information.
8. Add multiple images at the same time.

Here we can add our new items into the database.

Example :



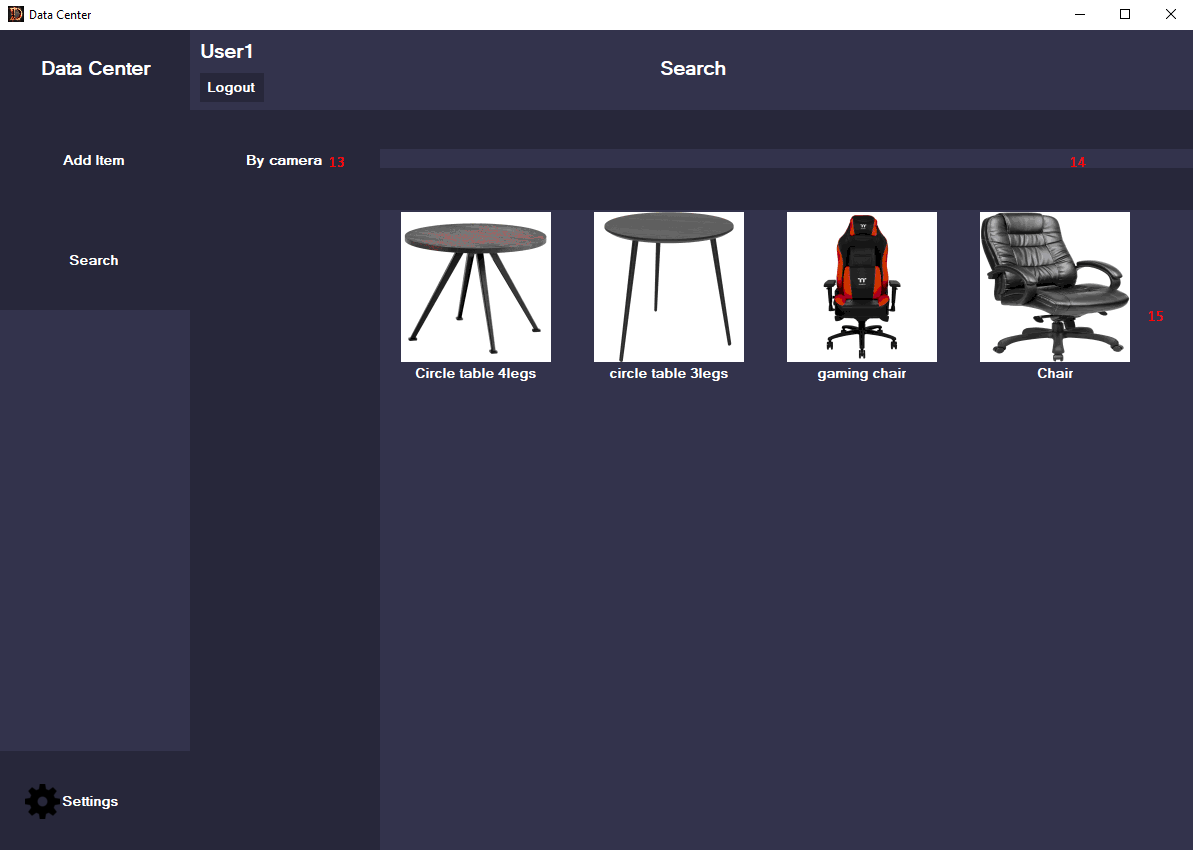
After inserting all the information you can press the “Save” button to process the items information and save it inside the database.

On success you will get a popup with the items barcode that you can scan to get information of the item.



* 1. **Search**

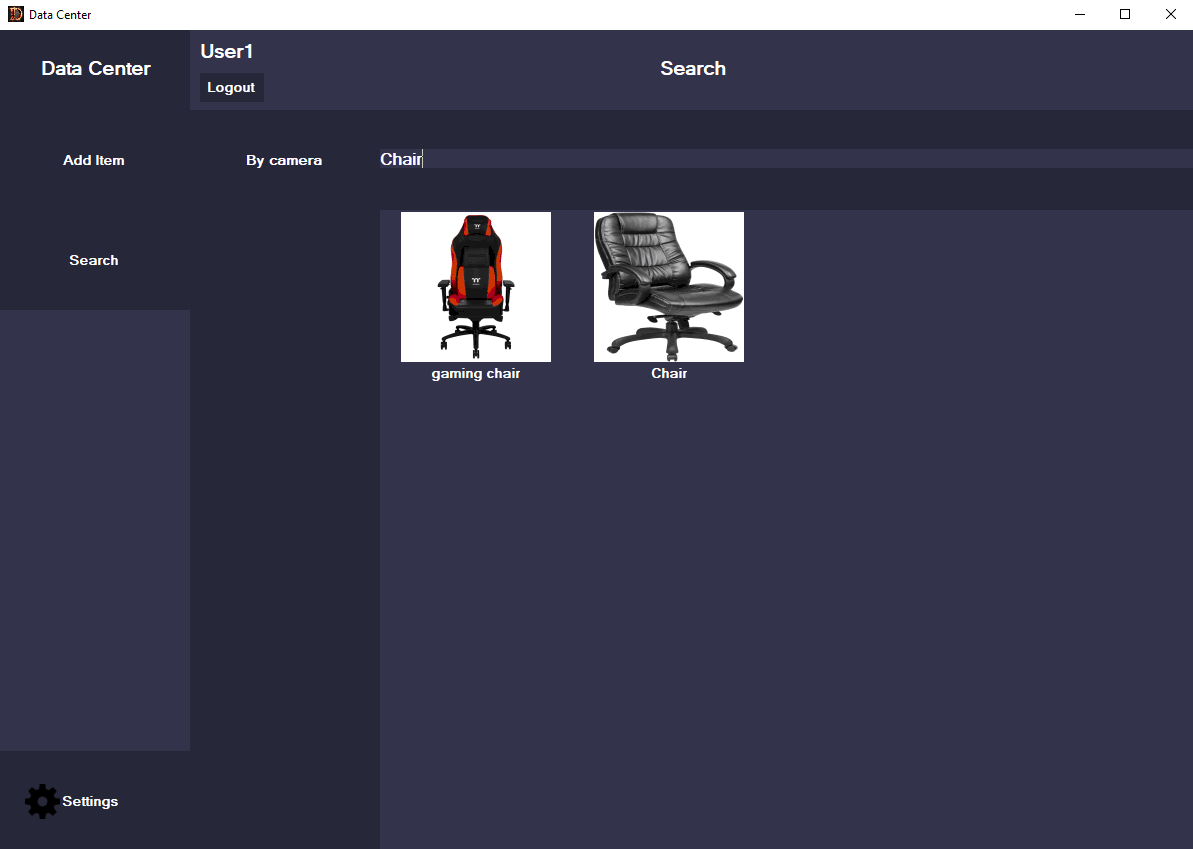
(4) On clicking the “Search” button you will be directed to a search form where you can search for the item via the search bar or by camera.



* + - 1. Switch to the camera to scan the barcode.
      2. Insert the name of the item your looking for.
      3. Display of all the items in the database.

Example:

Here we are looking for the Chair that we added on the previous example so on the search bar[14] we write “Chair”.



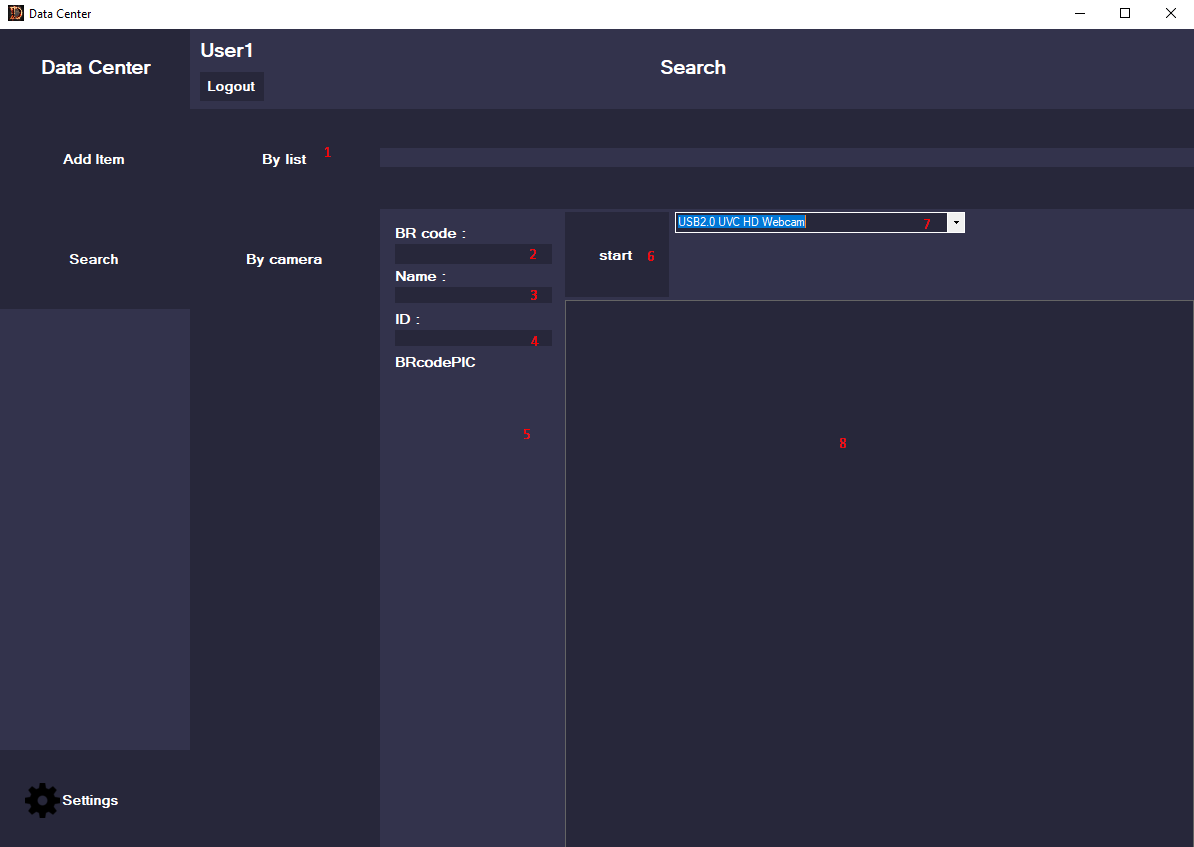
Here we get all the chairs in the database.

For more information about the item you just need to click it.

OR

By pressing the “By camera” button you will switch to camera page.

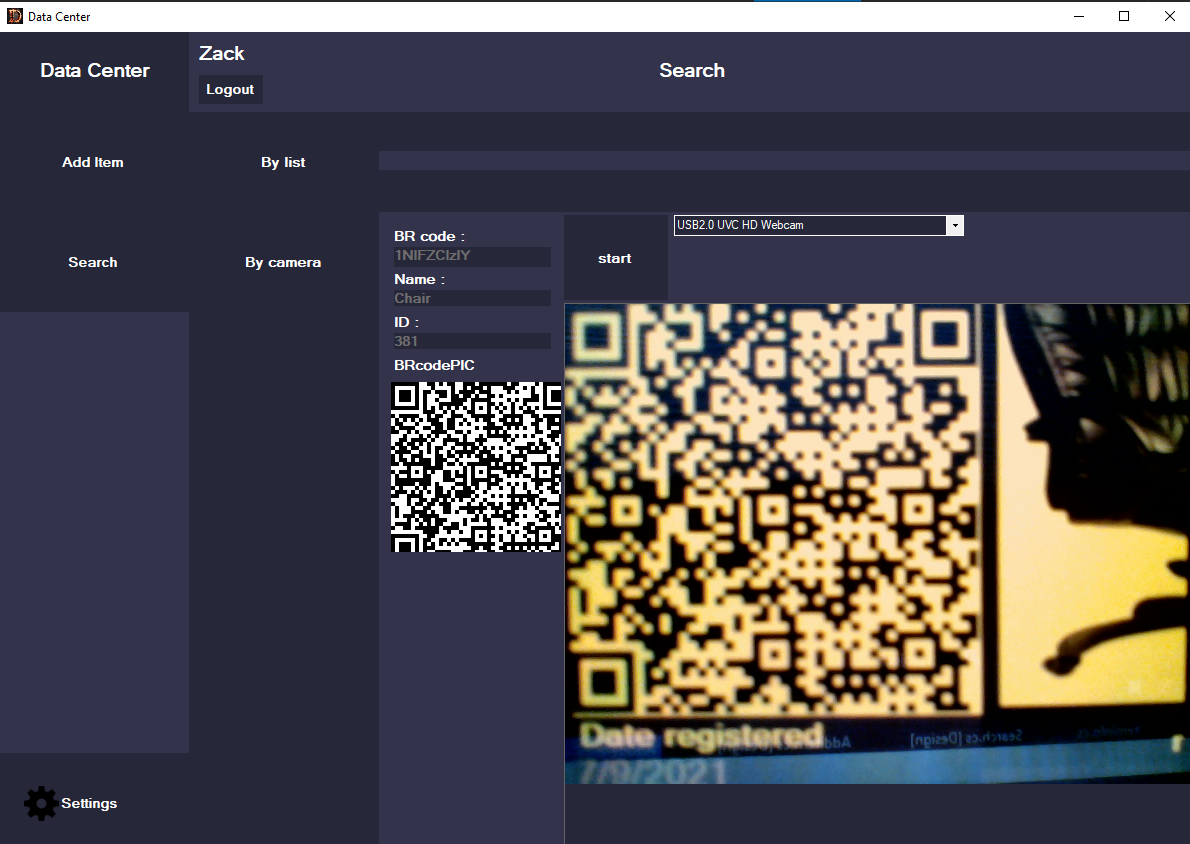
* 1. **Camera**



* + - 1. Go back to the list of all the items in the database.
      2. Barcodes unique string
      3. Name of the item
      4. ID of the item
      5. Barcode image
      6. Start camera
      7. List of all connected cameras to the computer.
      8. Display from the camera.

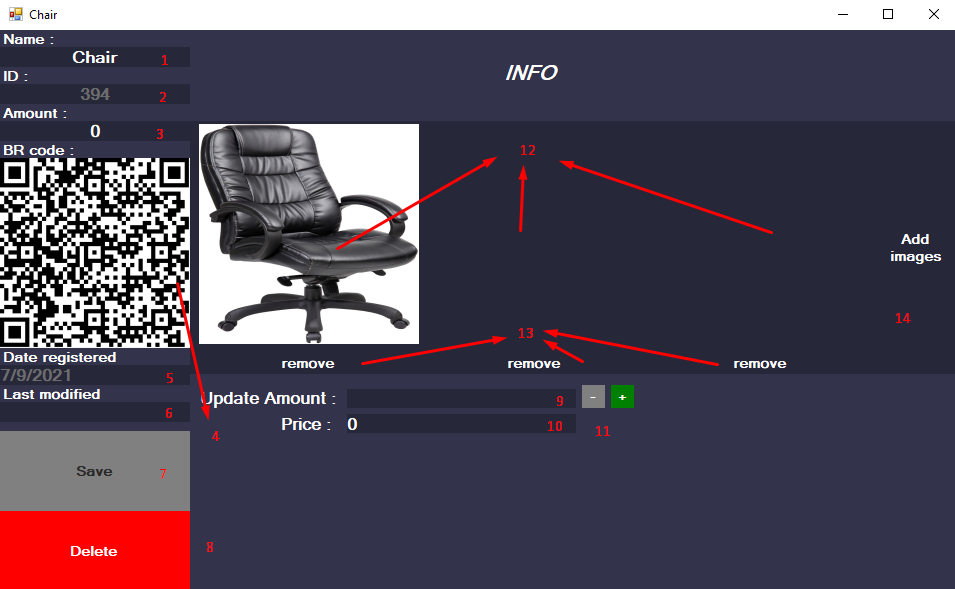
To read the Barcode all you need is to show the barcode to the camera for it to scan it on a successful scan you will automatically get all the items information.

A successful read you will get information about the item



And a pop up with More of the items information will show.

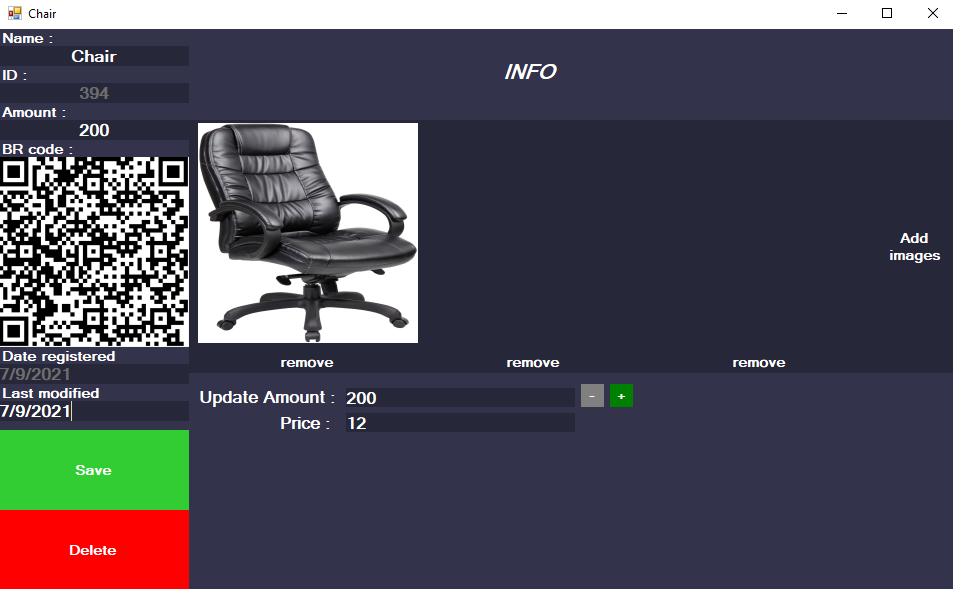
* 1. **Item information**



* + - 1. Name of the item.
      2. ID of the item.
      3. Quantity of the item.
      4. The unique barcode of the item.
      5. Date item was registered on.
      6. Last modification date of the item.
      7. Save changes.
      8. Delete item from database.
      9. Add or remove from the quantity.
      10. Price of the item per 1quantity.
      11. ADD/Subtract button.
      12. Clicking on the image box will allow you to place a new image.
      13. Remove image.
      14. Add multiple images

Example:

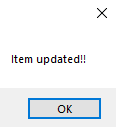
We decide to add 200 units of the chair so we click on “Update Amount” textbox[9] and write 200 and we give it a price of $12 USD.



The save button will glow green indicating that we can save any changes we made to the item.

Pressing Save will save all our changes.

On successful update you will get a popup.



Checking back on items information you will see that the amount[3] has updated from 0 to 200 and the last modification[6] date has been updated.

By clicking the “Last modification”[6] you can access a list with all the updates that the item has had.



Here you can see the dates when the changes happened the amounts

(original amount +/- user amount = total amount) and the price of the item at the time of updating information and by which user the items information was updated by.

Note:

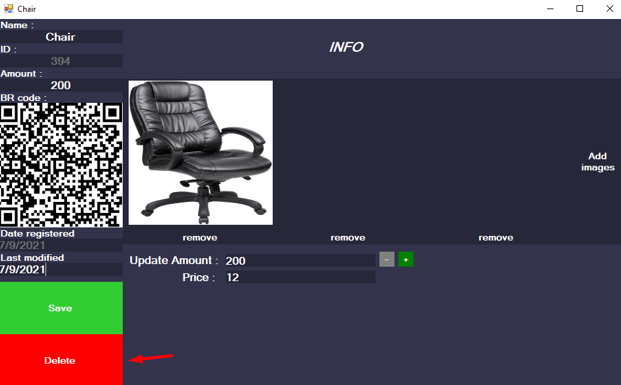
You can change the items pictures all except the barcode.

The Name of the item.

Amount to be added/subtracted.

* 1. **Delete**

If we want to delete the “Chair” we can do that by clicking the “Delete” button.

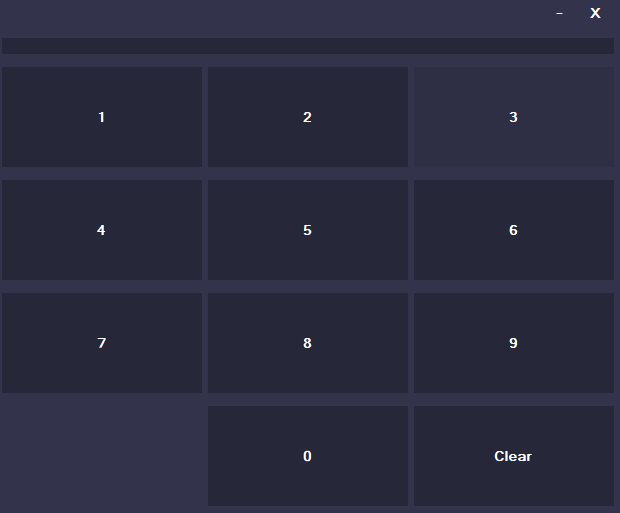


By pressing the “Delete” button the user will get a pop up requiring a passcode to delete an item.

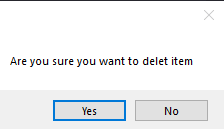
Note:

Not all account have a delete password meaning that they cant delete item .

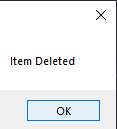
The delete code is different than the login in code.



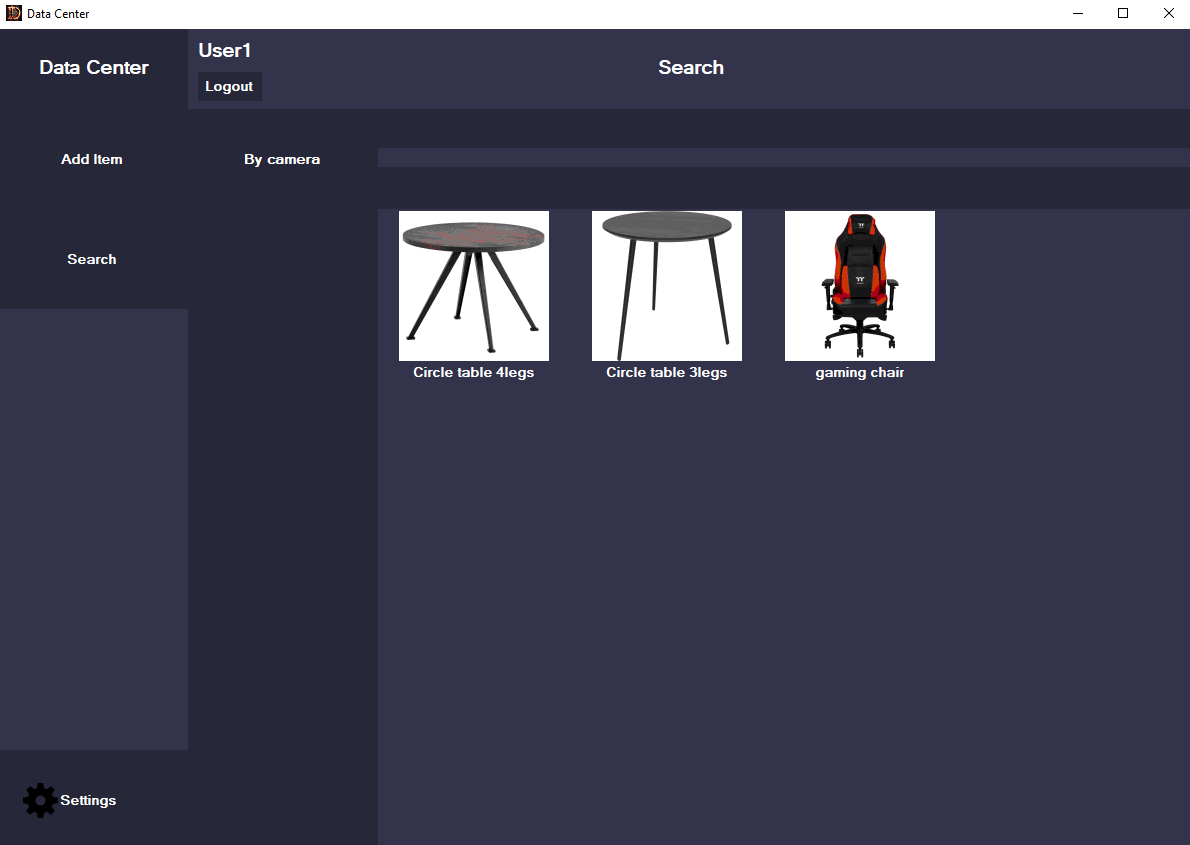
After typing the correct delete code you will get a final popup verification pressing yes will delete the item.



After Successfully deleting an item you will get a last popup telling you that the item got deleted .

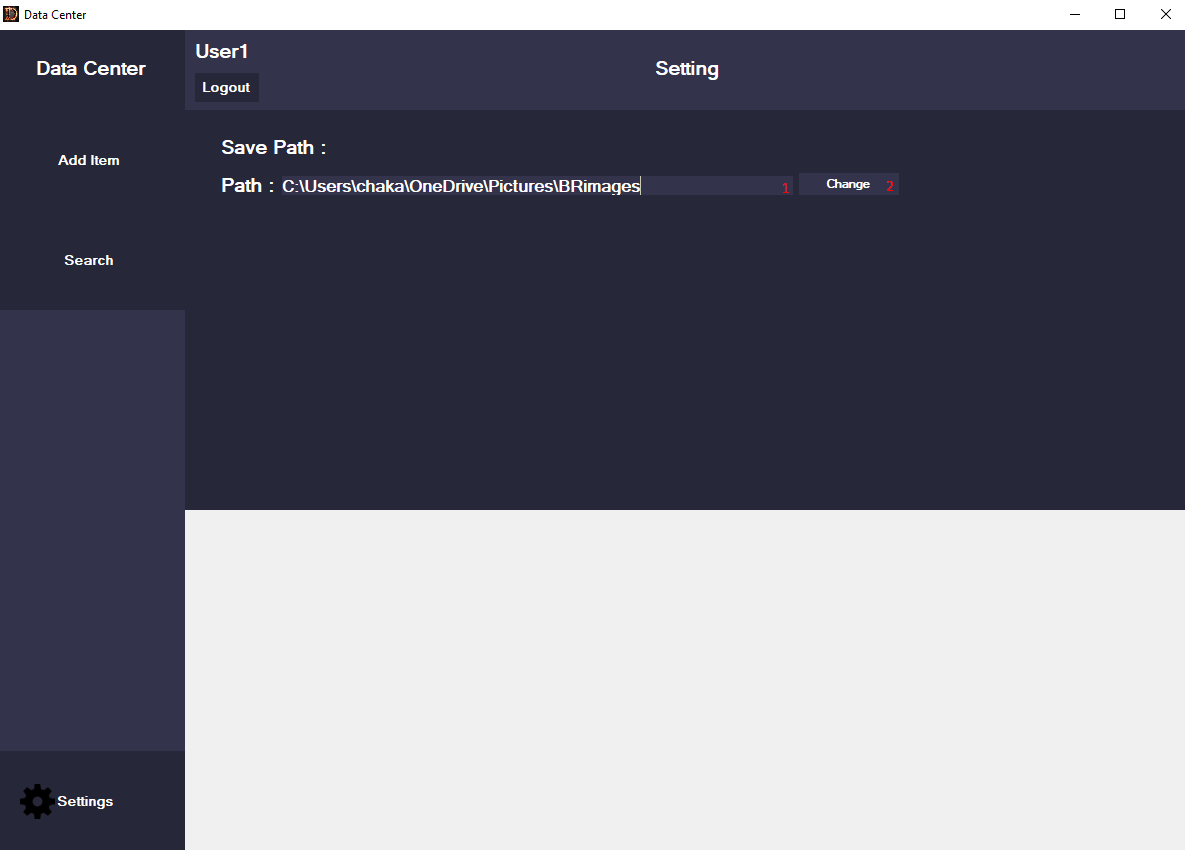


If we look back at the “Search” form we will see that our item is deleted from the database.



* 1. **Settings**

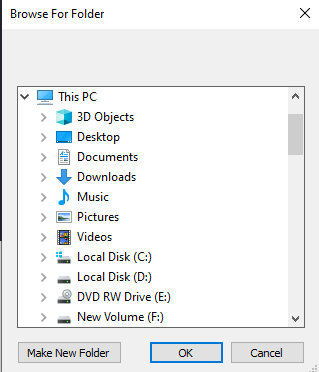
(5) Pressing the “Setting” button will take you to a page where you can change your barcode save path to another.



* + - 1. The location of the folder your barcode images are saved at.
      2. Changing the save location.

To change location all you need to do is press the change button.

You will get a popup asking you to choose a new save location.



If the path is valid then it will become the new default save location.

1. **Conclusion**

Using Software owners of warehouses can have an easier way keeping track of their products in storage whether its selling from an item or added more of the same item the software can help them keep track of them. By allowing employees in stores to access warehouse information with a click of a button, giving them a fast and reliable way of selling to customers while also updating the database so that other employees can see the changes in real time so no 2 employees can see the same item and end up selling more than there is in the warehouse to customers which can cause problems.

1. **References**

<https://dev.mysql.com/doc/dev/connector-net/6.10/html/T_MySql_Data_MySqlClient_MySqlDataReader.htm>

<https://www.tutorialspoint.com/what-do-column-flags-mean-in-mysql-workbench>