**Context**

1.Introduction……………………………………………………………………………………..3

2. Theoretical part............................................................................................................................4

3.Practical Part…………………………………………………………………………………….5

4.Conslusion……………………………………………………………………………………….8

5.Source code………………………………………………………………………………………9

**1.Introduction**

# The task of creating an automatic warehouse, which is assigned to the student, is to increase his level of practical skills.The system of "automated warehouse" is designed primarily to optimize the future costs of the customer, as the number of real employees in this type of warehouse is minimized.

The system was designed to provide clear and fast management of the warehouse, as well as its ease of use. And for a potential user, this means that it will be able to use the available service at any time, if necessary, contacting support.

The relevance of the project is that the owners of "automatic warehouses" are able to own / create many similar systems, complicate or, conversely, simplify their functionality.

**2. Theoretical part**

Description of requirements for the technical task:

• create a warehouse

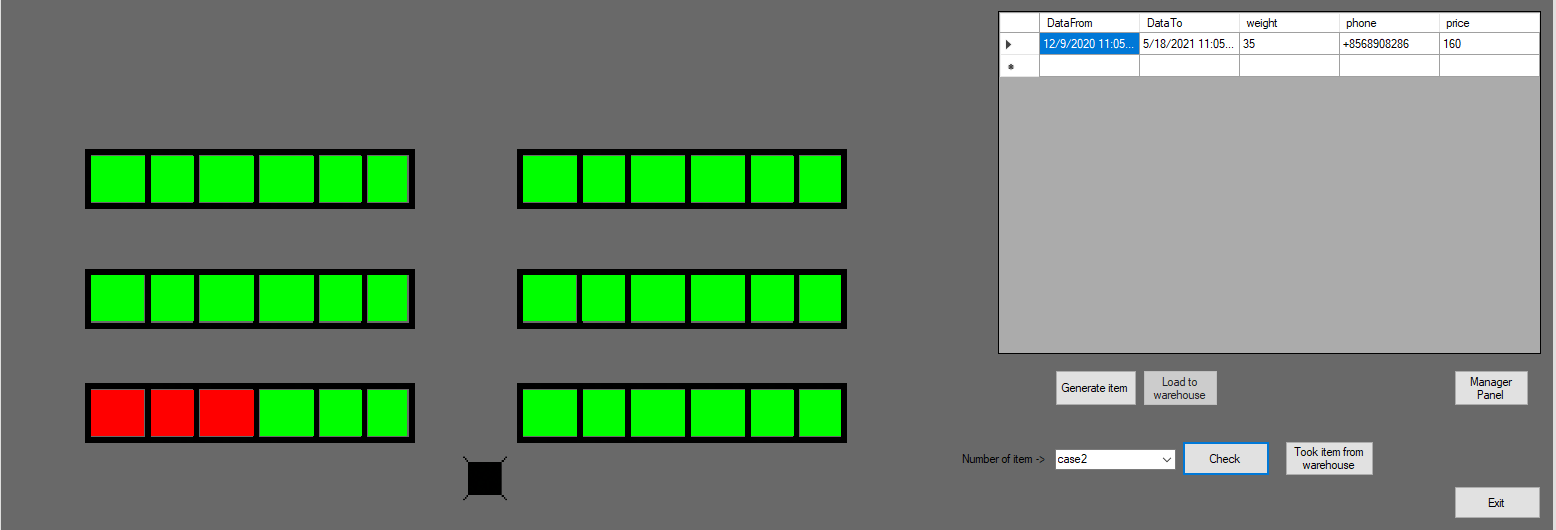
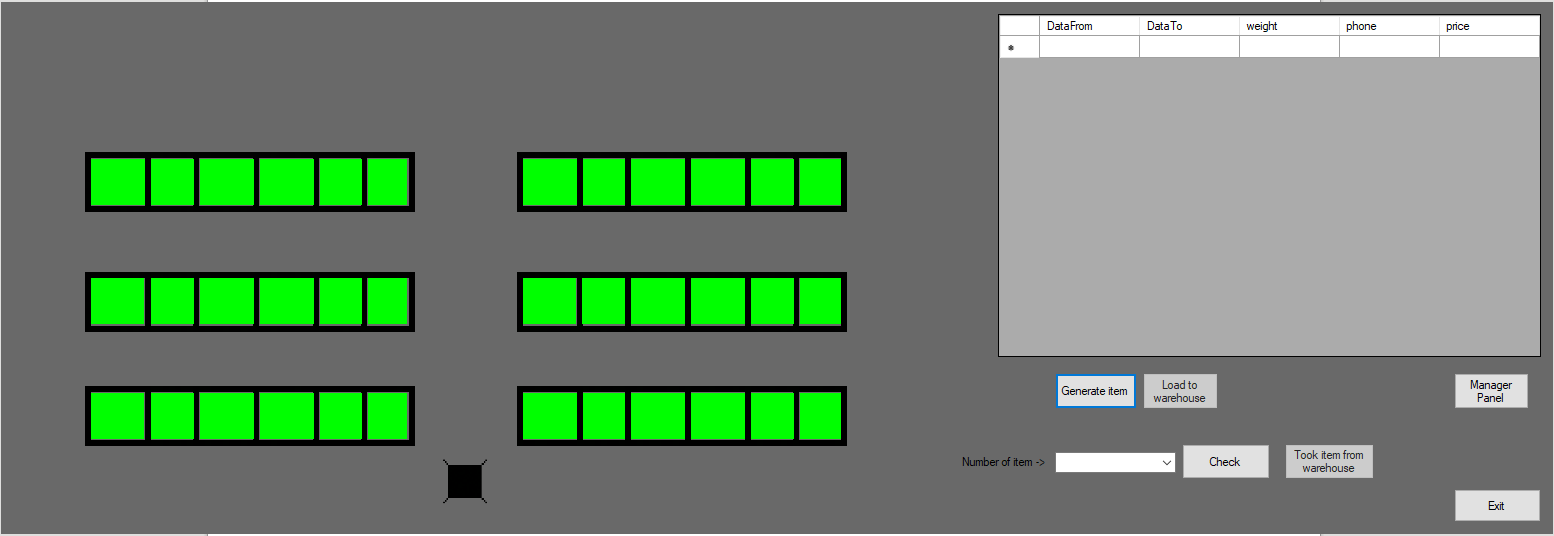
• to introduce an automatic system of filling and releasing warehouse cells (activity of a robot -loader)

• develop an automatic accounting management system

• create a user interface

• enable the manager to view all information about orders and analyze goods in stock

• introduce the possibility of autonomous operation of the system

**3.Practical Part**

3.1. Information about the warehouse:there are 36 cells in the warehouse - 12 in each row. The saving time is determined before loading to the warehouse, the payment is $ 1 per day and does not depend on the type of cargo. Payment is taken after loading. Free cells are highlighted in green,and busy are red.

3.2 Interface of interaction with the system

Description of buttons:

“Generate item” – creating cargo with random parameters such as date from what day we saving to date to what day we saving cargo, phone number, weight, and price.

“Load to warehouse” – saving cargo to database and moving object inside warehouse.

“Check” – checking cell which on empty or full from listbox from left side.

“Took item” – taking object to human selected in listbox.

And additional button for manager to give him possiblity to watch all database warehouse and correcting some info.

3.3.

Robot - loader - an automated maintenance mechanism that is able to fill and release the cells of the warehouse. The navigation system is carried out by magnetic labels located near each row and section. The load capacity of the robot is 300 kilograms. Before receiving the cargo or before returning it, the robot weighs it. And before returning, he waits until the customer makes the payment. The task of the robot is to transport goods, and it finds the necessary information about orders in the database of the system.

There may be situations when the customer will not be able to place an order for storage or receipt of goods in a timely manner. The following factors may prevent this:

-robbers

- no cargo

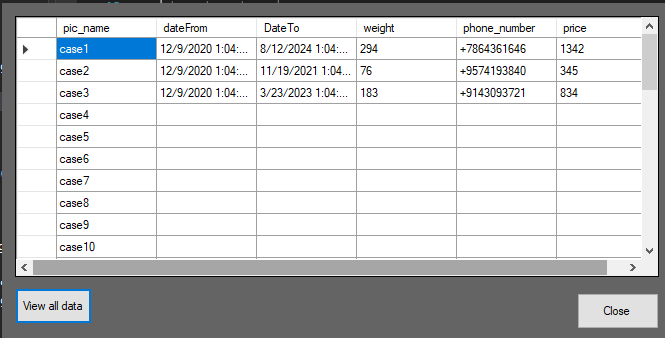
- the load "changed weight" when the robot weighed it before returning.

The probability of each of the above events is 0.5

When any problem occurs, the robot will immediately notify the system of the error, and the customer can contact support.

3.4 Interface of manager panel

The accounting system is database that implement formed / changed when creating orders by customers. The manager is able to view the entire database, and edit information in window interface.

****

**4.Conclusion**

According to the given technical task, it was possible to develop the scheme of a warehouse, to automate system of filling and release of warehouse cells, to develop automatic control system of the account, to create the convenient user interface.

Some emergency models have failed to implement.

**Source code**

The appendices contain program code written in С#

The database was created using sql lite.

using System;

using System.Data.SQLite;

using System.Drawing;

using System.Windows.Forms;

namespace warehouse

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

string cs = @"URI=file:C:\Users\Viper\source\repos\warehouse\bin\Debug\data.db";

Point stage0 = new Point(436, 442);

Point stage1 = new Point(438, 335);

Point stage2 = new Point(435, 217);

Point stage3 = new Point(436, 92);

public void Form1\_Paint(object sender, PaintEventArgs e, Keys KeyData)

{

pictureBox1.Image = Properties.Resources.pixil\_frame\_0\_\_1\_;

pictureBox2.Image = Properties.Resources.robot;

}

Random rnd = new Random();

Random gen = new Random();

int item\_weight;

int range;

DateTime randomDate;

DateTime ourDays;

string phone\_number;

int price;

public void button1\_Click(object sender, EventArgs e)

{

dataGridView1.Rows.Clear();

item\_weight = rnd.Next(1, 400);

range = 5 \* 365;

ourDays = DateTime.Now;

randomDate = ourDays.AddDays(gen.Next(2, range));

phone\_number = GetRandomTelNo();

price = (int)(randomDate - ourDays).TotalDays; ;

if (item\_weight > 300)

{

MessageBox.Show("I can't took this weight,call human", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (item\_weight < 300)

{

dataGridView1.Rows.Add(new string[] { ourDays.ToString(), randomDate.ToString(), item\_weight.ToString(), phone\_number, price.ToString() });

//= String.Format("Saving to date {0} ,weight is {1} kg", randomDate, item\_weight);

button2.Enabled = true;

}

}

static string GetRandomTelNo()

{

var random = new Random();

string s = String.Empty;

char a = (char)43;

for (int i = 0; i < 10; i++)

s = String.Concat(s, random.Next(10).ToString());

return s = String.Concat(a, s);

}

public PictureBox getPictureBoxByName(string name)

{

foreach (object p in this.Controls)

{

if (p.GetType() == typeof(PictureBox))

if (((PictureBox)p).Name == name)

return (PictureBox)p;

}

return new PictureBox(); //OR return null;

}

string pic\_number = String.Empty;

public void button2\_Click\_1(object sender, EventArgs e)

{

button2.Enabled = false;

dataGridView1.Rows.Clear();

using var con = new SQLiteConnection(cs);

con.Open();

using var cmd = new SQLiteCommand(con);

string line = String.Empty;

cmd.CommandText = "SELECT pic\_name, CASE WHEN is\_full = 0 THEN 'EMPTY' END as 'DATA' FROM data;";

SQLiteDataReader r = cmd.ExecuteReader();

while (r.Read())

{

pic\_number = r["pic\_name"].ToString();

line = r["DATA"].ToString();

if (line == "EMPTY")

{

break;

}

}

r.Close();

temp = getPictureBoxByName(pic\_number);

if (temp.Location.X > stage1.X)

{

Xdirection = 1;

Sprava = false;

}

else

{

Xdirection = -1;

Sprava = true;

}

if (temp.Location.Y > stage1.Y)

{

height = stage1.Y;

}

else if (temp.Location.Y > stage2.Y && temp.Location.Y < stage1.Y)

{

height = stage2.Y;

}

else //(temp.Location.Y<stage2.Y && temp.Location.Y>stage3.Y)

{

height = stage3.Y;

}

cmd.CommandText = "UPDATE data SET is\_full=1,dateFrom= :dateFrom,DateTo= :DateTo ,weight= :weight,phone\_number= :phone\_number,price= :price WHERE pic\_name = :pic\_number";

cmd.Parameters.AddWithValue("dateFrom", ourDays.ToString());

cmd.Parameters.AddWithValue("DateTo", randomDate.ToString());

cmd.Parameters.AddWithValue("weight", item\_weight);

cmd.Parameters.AddWithValue("pic\_number", pic\_number);

cmd.Parameters.AddWithValue("phone\_number", phone\_number);

cmd.Parameters.AddWithValue("price", Math.Abs(price));

cmd.ExecuteNonQuery();

Task = true;

MoveTimer.Start();

}

public void Form1\_Load(object sender, EventArgs e)

{

// using var con = new SQLiteConnection(cs);

// con.Open();

// using var cmd = new SQLiteCommand(con);

//// cmd.CommandText = "DROP TABLE IF EXISTS data";

// //cmd.ExecuteNonQuery();

// //cmd.CommandText = @"CREATE TABLE data(id INTEGER PRIMARY KEY, pic\_name TEXT,is\_full INTEGER NOT NULL DEFAULT 0 CHECK(is\_full IN (0,1)),date TEXT, weight INT)";

// //cmd.ExecuteNonQuery();

// //var connection = new SQLiteConnection("Data Source=:memory:");

// foreach (var pb in this.Controls.OfType<PictureBox>())

// {

// //cmd.CommandText = "INSERT INTO data(pic\_name,is\_full) VALUES(@pic\_name,@is\_full)";

// // cmd.Parameters.AddWithValue("@pic\_name", pb.Name);

// //cmd.Parameters.AddWithValue("@is\_full", 0);

// //cmd.ExecuteNonQuery();

//

// }

}

private void button3\_Click(object sender, EventArgs e)

{

using var con = new SQLiteConnection(cs);

con.Open();

using var cmd = new SQLiteCommand(con);

cmd.CommandText = @"UPDATE data SET is\_full=0,dateFrom=NULL,DateTo=NULL,weight=NULL,phone\_number=NULL,price=NULL";

cmd.ExecuteNonQuery();

con.Close();

Application.Exit();

}

private void button5\_Click(object sender, EventArgs e)

{

dataGridView1.Rows.Clear();

using var con = new SQLiteConnection(cs);

con.Open();

using var cmd = new SQLiteCommand(con);

string weight = String.Empty;

string dateFrom = String.Empty;

string dateTo = String.Empty;

string is\_full = String.Empty;

string pic\_name = String.Empty;

string phone\_number = String.Empty;

string price = String.Empty;

cmd.CommandText = "SELECT \* FROM data WHERE pic\_name= :pic\_name";

cmd.Parameters.AddWithValue("pic\_name", comboBox1.Text);

SQLiteDataReader r = cmd.ExecuteReader();

while (r.Read())

{

pic\_name = r["pic\_name"].ToString();

is\_full = r["is\_full"].ToString();

dateFrom = r["dateFrom"].ToString();

dateTo = r["DateTo"].ToString();

weight = r["weight"].ToString();

phone\_number = r["phone\_number"].ToString();

price = r["price"].ToString();

if (is\_full == "1")

{

MessageBox.Show("FULL", comboBox1.Text);

button4.Enabled = true;

}

else

{

MessageBox.Show("Empty", comboBox1.Text);

}

}

dataGridView1.Rows.Add(new string[] { dateFrom, dateTo, weight, phone\_number, price });

r.Close();

con.Close();

}

private void button4\_Click(object sender, EventArgs e)

{

button4.Enabled = false;

dataGridView1.Rows.Clear();

using var con = new SQLiteConnection(cs);

con.Open();

using var cmd = new SQLiteCommand(con);

cmd.CommandText = "UPDATE data SET is\_full = 0, dateFrom = NULL,DateTo=NULL, weight = NULL,phone\_number=NULL,price=NULL WHERE pic\_name = :pic\_name";

cmd.Parameters.AddWithValue("pic\_name", comboBox1.Text);

cmd.ExecuteNonQuery();

temp = getPictureBoxByName(comboBox1.Text);

if (temp.Location.X > stage1.X)

{

Xdirection = 1;

Sprava = false;

}

else

{

Xdirection = -1;

Sprava = true;

}

if (temp.Location.Y > stage1.Y)

{

height = stage1.Y;

}

else if (temp.Location.Y > stage2.Y && temp.Location.Y < stage1.Y)

{

height = stage2.Y;

}

else

{

height = stage3.Y;

}

Task = true;

}

int speed = 20;

PictureBox temp;

bool iSOnPlace = false;

int Xdirection = 0;

int Ydirection = -1;

bool Sprava = false;

bool Task = false;

int height = 335;

private void MoveTimer\_Tick(object sender, EventArgs e)

{

if (Task)

{

if (Sprava)

{

if (!iSOnPlace)

{

if (pictureBox2.Location.Y > height)

{

pictureBox2.Location = new Point(pictureBox2.Location.X, pictureBox2.Location.Y + speed \* Ydirection);

}

else

{

if (pictureBox2.Location.X > temp.Location.X)

{

pictureBox2.Location = new Point(pictureBox2.Location.X + speed \* Xdirection, pictureBox2.Location.Y);

}

else

{

iSOnPlace = true;

if (temp.BackColor == Color.Lime)

{

temp.BackColor = Color.Red;

}

else //if(temp.BackColor == Color.Red)

{

temp.BackColor = Color.Lime;

}

Ydirection = -Ydirection;

Xdirection = -Xdirection;

}

}

}

else

{

if (pictureBox2.Location.X < stage1.X)

{

pictureBox2.Location = new Point(pictureBox2.Location.X + speed \* Xdirection, pictureBox2.Location.Y);

}

else

{

if (pictureBox2.Location.Y < stage0.Y)

{

pictureBox2.Location = new Point(pictureBox2.Location.X, pictureBox2.Location.Y + Ydirection \* speed);

}

else

{

iSOnPlace = false;

Task = false;

Ydirection = -Ydirection;

}

}

}

}

else

{

if (!iSOnPlace)

{

if (pictureBox2.Location.Y > height)

{

pictureBox2.Location = new Point(pictureBox2.Location.X, pictureBox2.Location.Y + speed \* Ydirection);

}

else

{

if (pictureBox2.Location.X < temp.Location.X)

{

pictureBox2.Location = new Point(pictureBox2.Location.X + speed \* Xdirection, pictureBox2.Location.Y);

}

else

{

iSOnPlace = true;

if (temp.BackColor == Color.Lime)

{

temp.BackColor = Color.Red;

}

else //if(temp.BackColor == Color.Red)

{

temp.BackColor = Color.Lime;

}

Ydirection = -Ydirection;

Xdirection = -Xdirection;

}

}

}

else

{

if (pictureBox2.Location.X > stage1.X)

{

pictureBox2.Location = new Point(pictureBox2.Location.X + speed \* Xdirection, pictureBox2.Location.Y);

}

else

{

if (pictureBox2.Location.Y < stage0.Y)

{

pictureBox2.Location = new Point(pictureBox2.Location.X, pictureBox2.Location.Y + Ydirection \* speed);

}

else

{

iSOnPlace = false;

Task = false;

Ydirection = -Ydirection;

}

}

}

}

}

}

private void button6\_Click(object sender, EventArgs e)

{

ManagerPanel mg = new ManagerPanel();

mg.Show();

}

}

}

Manager Panel

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SQLite;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace warehouse

{

public partial class ManagerPanel : Form

{

public ManagerPanel()

{

InitializeComponent();

}

string cs = @"URI=file:C:\Users\Viper\source\repos\warehouse\bin\Debug\data.db";

private void button1\_Click(object sender, EventArgs e)

{

dataGridView1.Rows.Clear();

using var con = new SQLiteConnection(cs);

con.Open();

using var cmd = new SQLiteCommand(con);

string pic\_name = String.Empty;

string dateFrom = String.Empty;

string DateTo = String.Empty;

string weight = String.Empty;

string phone\_number = String.Empty;

string price = String.Empty;

cmd.CommandText = "SELECT pic\_name,dateFrom,DateTo,weight,phone\_number,price FROM data;";

cmd.ExecuteNonQuery();

SQLiteDataReader r = cmd.ExecuteReader();

while (r.Read())

{

pic\_name = r["pic\_name"].ToString();

dateFrom = r["dateFrom"].ToString();

DateTo = r["DateTo"].ToString();

weight = r["weight"].ToString();

phone\_number = r["phone\_number"].ToString();

price = r["price"].ToString();

dataGridView1.Rows.Add(new string[] { pic\_name, dateFrom, DateTo, weight, phone\_number, price });

}

r.Close();

con.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}