Restaurant Manager

Cerința

Creati un serviciu Web care ofera acces la o baza de date care stocheaza informatii. Apoi,

creati un client de tip Windows Form App care utilizeaza serviciul. Serviciul va oferi metode de

adaugarea si modificare a informatiilor. Tema aplicatiei este la alegere.

Rezolvare

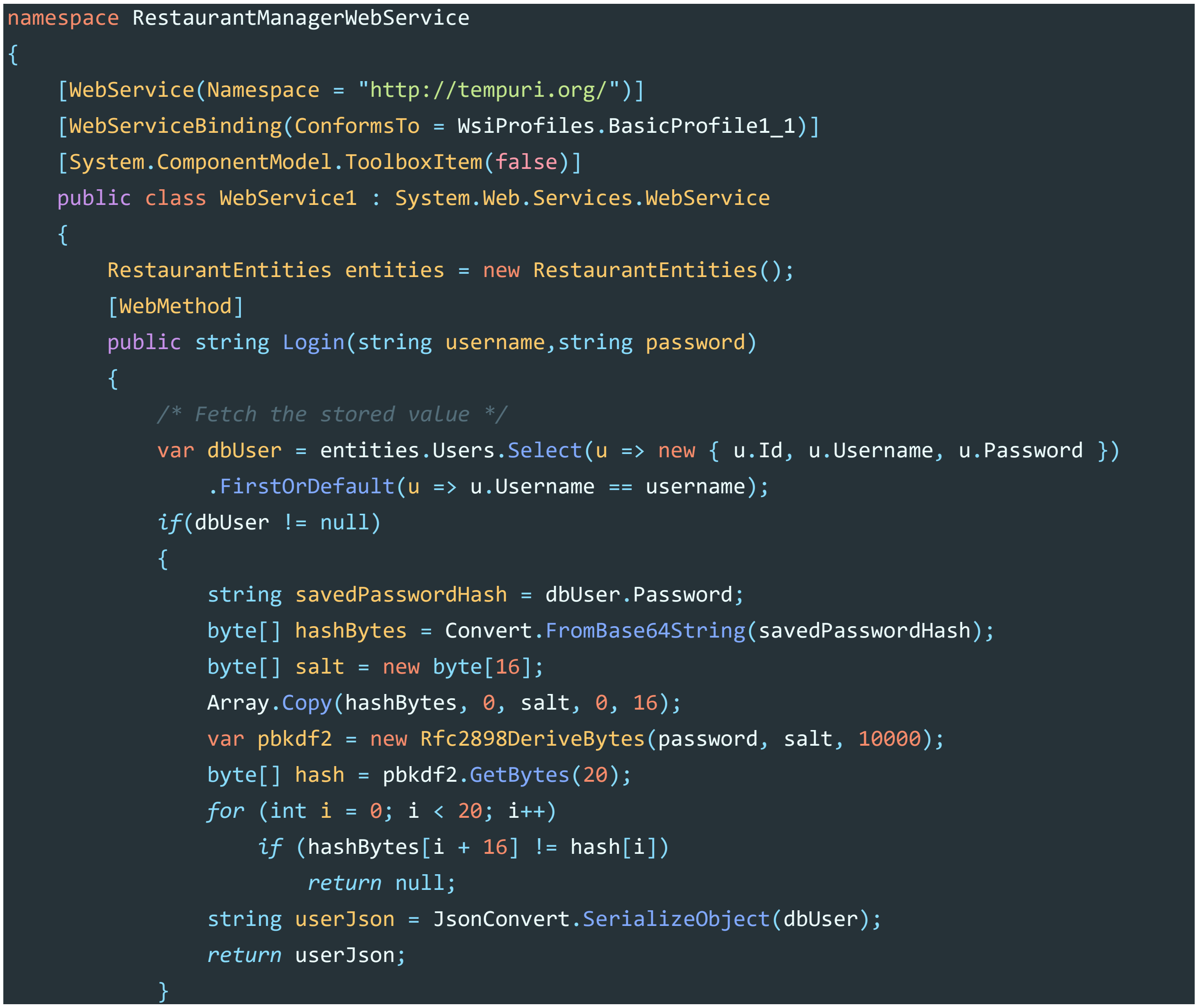
Pentru tema acestei cerințe am construit o aplicație a cărei rol este să ajute restaurantele să țină evidența rezervărilor printr-un sistem centralizat, care permite stocarea rezervărilor și totodată oferă o gestionare mai ușoară a meselor restaurantului.

Prezentare Aplicație

1.Webservice

Toate operațiunile de Create,Read,Update și Delete sunt realizate prin intermediul Serviciului Web.

Pe partea de serviciu web am utilizat libraria externa Web.Api și mai exact packetul Newtonsoft.Json pentru capacitatea de a serializa obiecte.



```
namespace RestaurantManagerWebService
{
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    public class WebService1 : System.Web.Services.WebService
    {
        RestaurantEntities entities = new RestaurantEntities();
        [WebMethod]
        public string Login(string username,string password)
        {
            /* Fetch the stored value */
            var dbUser = entities.Users.Select(u => new { u.Id, u.Username, u.Password })
                .FirstOrDefault(u => u.Username == username);
            if(dbUser != null)
            {
                string savedPasswordHash = dbUser.Password;
                byte[] hashBytes = Convert.FromBase64String(savedPasswordHash);
                byte[] salt = new byte[16];
                Array.Copy(hashBytes, 0, salt, 0, 16);
                var pbkdf2 = new Rfc2898DeriveBytes(password, salt, 10000);
                byte[] hash = pbkdf2.GetBytes(20);
                for (int i = 0; i < 20; i++)
                    if (hashBytes[i + 16] != hash[i])
                        return null;
                string userJson = JsonConvert.SerializeObject(dbUser);
                return userJson;
            }
            return null;
        }
        public string hashPassword(string password)
        {
            byte[] salt;
            new RNGCryptoServiceProvider().GetBytes(salt = new byte[16]);
            var pbkdf2 = new Rfc2898DeriveBytes(password, salt, 10000);
            byte[] hash = pbkdf2.GetBytes(20);
            byte[] hashBytes = new byte[36];
            Array.Copy(salt, 0, hashBytes, 0, 16);
            Array.Copy(hash, 0, hashBytes, 16, 20);
            string savedPasswordHash = Convert.ToBase64String(hashBytes);
            return savedPasswordHash;
        }
        [WebMethod]
        public string getReservations(int id)
        {
            var dbReservations = entities.Reservations
                .Where(r => r.RestaurantId == id)
                .Select(r => new { r.ReservationId, r.ReservationDate, r.ReservationTime, r.PartySize, r.ReservationDetails, r.ClientId, r.Client.FirstName, r.Client.LastName,r.Client.PhoneNumber });
            string reservationJson = JsonConvert.SerializeObject(dbReservations);
            return reservationJson;
        }
        [WebMethod]
        public void deleteReservation (int id)
        {
            var reservation = entities.Reservations.Find(id);
            entities.Entry(reservation).State = EntityState.Deleted;
            entities.SaveChanges();
        }
        [WebMethod]
        public void addReservation(string Firstname,string Lastname,string Phonenumber,int PartySize,DateTime Date,int Hour,int Minute,string Description,int RestaurantId)
        {
            Client client = entities.Clients.FirstOrDefault(c => c.RestaurantId == RestaurantId && c.FirstName == Firstname && c.LastName == Lastname && c.PhoneNumber == Phonenumber);
            Reservation reservation = new Reservation();
            if (client is null)
            {
                client = new Client();
                client.FirstName = Firstname;
                client.LastName = Lastname;
                client.PhoneNumber = Phonenumber;
                client.RestaurantId = RestaurantId;
                entities.Clients.Add(client);
                entities.SaveChanges();
            }
            TimeSpan Time = new TimeSpan(Hour, Minute, 0);
            reservation.Client = client;
            reservation.PartySize = PartySize;
            reservation.ReservationDate = Date;
            reservation.ReservationTime = Time;
            reservation.ReservationDetails = Description;
            reservation.RestaurantId = RestaurantId;
            entities.Reservations.Add(reservation);
            entities.SaveChanges();
        }
        [WebMethod]
        public void editReservation(int ReservationId,int clientId,string Firstname, string Lastname, string Phonenumber, int PartySize, DateTime Date, int Hour, int Minute, string Description, int RestaurantId)
        {
            Client client = entities.Clients.FirstOrDefault(c => c.ClientId == clientId);
            Reservation reservation = entities.Reservations.FirstOrDefault(r => r.ReservationId == ReservationId);
            client.FirstName = Firstname;
            client.LastName = Lastname;
            client.PhoneNumber = Phonenumber;
            client.RestaurantId = RestaurantId;
            TimeSpan Time = new TimeSpan(Hour, Minute, 0);
            reservation.Client = client;
            reservation.PartySize = PartySize;
            reservation.ReservationDate = Date;
            reservation.ReservationTime = Time;
            reservation.ReservationDetails = Description;
            entities.SaveChanges();
        }
    }
}
```

Client Windows Forms:

Pe partea de windows forms vom prezenta mai întâi modelele.

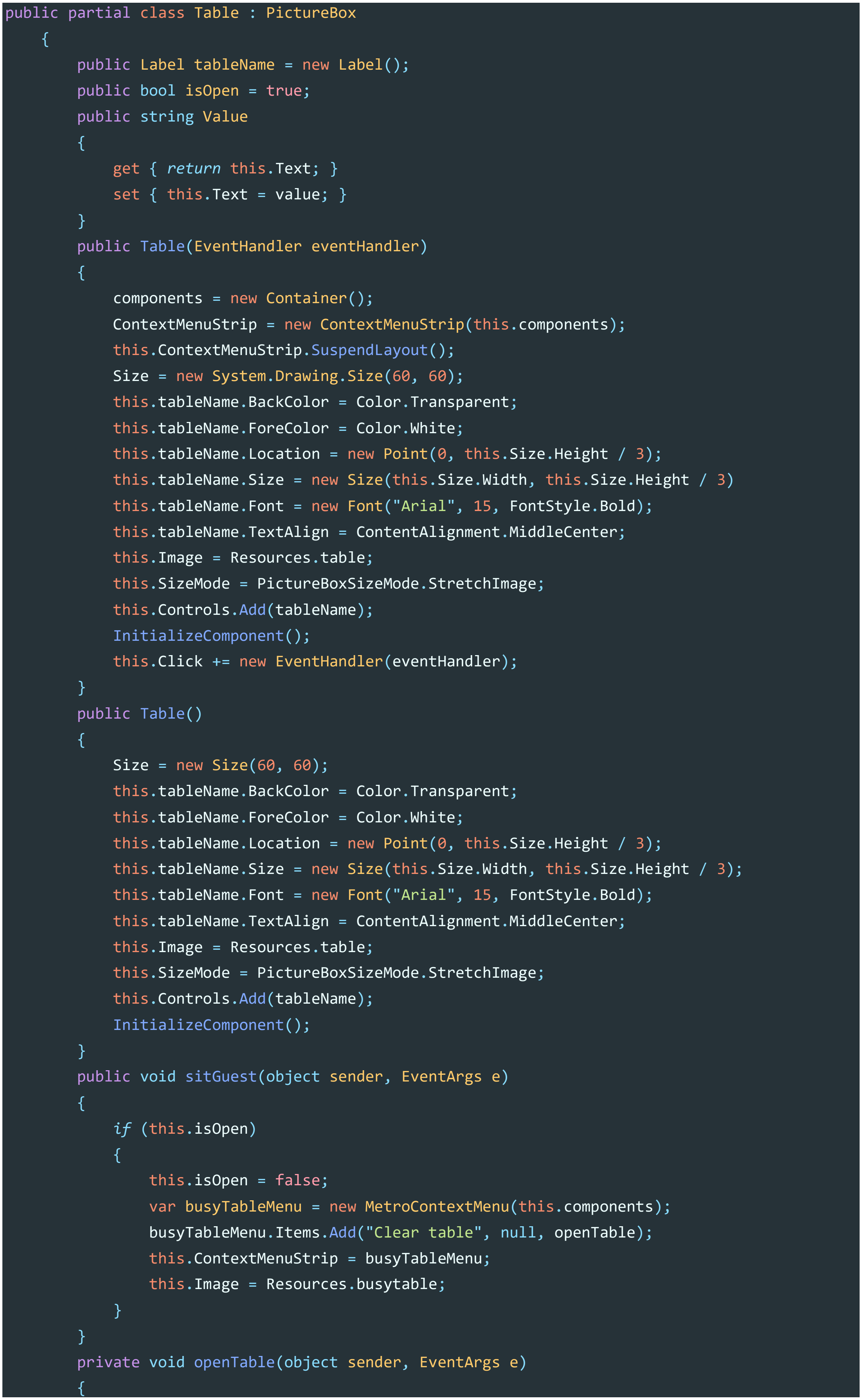
Clasa User

```
 public class User
    {
        public int Id { get; set; }
        public string Username { get; set; }
        public string Password { get; set; }
        List<Reservation> reservations { get; set; }
    }
```

Clasa Reservations

```
public class Reservation
    {
        public int ReservationId;
        public DateTime ReservationDate;
        public TimeSpan ReservationTime;
        public int PartySize;
        public string ReservationDetails;
        public int ClientId;
        public string FirstName;
        public string LastName;
        public string PhoneNumber;
    }
```

Custom Control : Table



```
public partial class Table : PictureBox
    {
        public Label tableName = new Label();
        public bool isOpen = true;
        public string Value
        {
            get { return this.Text; }
            set { this.Text = value; }
        }
        public Table(EventHandler eventHandler)
        {
            components = new Container();
            ContextMenuStrip = new ContextMenuStrip(this.components);
            this.ContextMenuStrip.SuspendLayout();
            Size = new System.Drawing.Size(60, 60);
            this.tableName.BackColor = Color.Transparent;
            this.tableName.ForeColor = Color.White;
            this.tableName.Location = new Point(0, this.Size.Height / 3);
            this.tableName.Size = new Size(this.Size.Width, this.Size.Height / 3)
            this.tableName.Font = new Font("Arial", 15, FontStyle.Bold);
            this.tableName.TextAlign = ContentAlignment.MiddleCenter;
            this.Image = Resources.table;
            this.SizeMode = PictureBoxSizeMode.StretchImage;
            this.Controls.Add(tableName);
            InitializeComponent();
            this.Click += new EventHandler(eventHandler);
        }
        public Table()
        {
            Size = new Size(60, 60);
            this.tableName.BackColor = Color.Transparent;
            this.tableName.ForeColor = Color.White;
            this.tableName.Location = new Point(0, this.Size.Height / 3);
            this.tableName.Size = new Size(this.Size.Width, this.Size.Height / 3);
            this.tableName.Font = new Font("Arial", 15, FontStyle.Bold);
            this.tableName.TextAlign = ContentAlignment.MiddleCenter;
            this.Image = Resources.table;
            this.SizeMode = PictureBoxSizeMode.StretchImage;
            this.Controls.Add(tableName);
            InitializeComponent();
        }
        public void sitGuest(object sender, EventArgs e)
        {
            if (this.isOpen)
            {
                this.isOpen = false;
                var busyTableMenu = new MetroContextMenu(this.components);
                busyTableMenu.Items.Add("Clear table", null, openTable);
                this.ContextMenuStrip = busyTableMenu;
                this.Image = Resources.busytable;
            }
        }
        private void openTable(object sender, EventArgs e)
        {
            if (!this.isOpen)
            {
                this.Image = Resources.table;
                this.isOpen = true;
                var tableMenu = new MetroContextMenu(this.components);
                tableMenu.Items.Add("Add guest", null, sitGuest);
                this.ContextMenuStrip = tableMenu;
            }
        }
        protected override void OnPaint(PaintEventArgs pe)
        {
            base.OnPaint(pe);
            try
            {
                this.tableName.Text = Tag.ToString();
            }
            catch (Exception e) { }
        }
    }
```

Forms

Loginform

```
public partial class LoginForm : MetroFramework.Forms.MetroForm
    {
        WebService1SoapClient service = new WebService1SoapClient();
        public LoginForm()
        {
            InitializeComponent();
        }
        private void loginButton_Click(object sender, EventArgs e)
        {
            loginSpinner.Visible = true;
            string username = this.usernameTextbox.Text;
            string password = this.passwordTextbox.Text;
            string serviceResponse = service.Login(username, password);
            if (serviceResponse != null)
            {
                loginSpinner.Visible = false;
                User user = JsonConvert.DeserializeObject<User>(serviceResponse);
                RestaurantForm restaurantForm = new RestaurantForm(user,service);
                restaurantForm.Show();
                this.Hide();
            }
            else
            {
                statusLabel.Text = "Wrong Username or Password";
                statusLabel.ForeColor = Color.Red;
                statusLabel.Visible = true;
                loginSpinner.Visible = false;
            }
        }
    }
```

RestaurantForm

```
    public partial class RestaurantForm : MetroFramework.Forms.MetroForm
    {
        public WebService1SoapClient service;
        User user;
        List<Reservation> reservations;
        public RestaurantForm(User user, WebService1SoapClient service)
        {
            InitializeComponent();
            this.user = user;
            this.service = service;
            refreshReservationsDB();
        }
        public void refreshReservationsDB()
        {

         var reservationResponse = this.service.getReservations(this.user.Id);
         reservations = JsonConvert.DeserializeObject<List<Reservation>>(reservationResponse);
         refreshReservations();
        }
        public void refreshReservations()
        {
            this.reservationListbox.Items.Clear();
            reservationListbox.ValueMember = "ReservationId";
            reservationListbox.DisplayMember = "Text";
            reservations.ForEach(r => {
                if (r.ReservationDate.ToString().Split(' ')[0].Equals(datePicker.Value.ToString().Split(' ')[0]))
                    reservationListbox.Items.Add(new { r.ReservationId, Text = "[" + r.PartySize.ToString() + "] " + r.FirstName.ToString() + " " + r.LastName.ToString() });
            });
        }
        private void reservationButton_Click(object sender, EventArgs e)
        {
            ReservationForm reservationForm = new ReservationForm(user,this);
            reservationForm.ShowDialog();
        }
        void reservationListbox_MouseDoubleClick(object sender, MouseEventArgs e)
        {
            int index = this.reservationListbox.IndexFromPoint(e.Location);
            if (index != System.Windows.Forms.ListBox.NoMatches)
            {
                int reservationId = Convert.ToInt32(reservationListbox.Items[index].ToString().Split(',')[0].Split('=')[1]);
                Reservation reservation = reservations.FirstOrDefault(r => r.ReservationId == reservationId);
                new ReservationForm(reservation, user,this).Show();

            }
        }
        private void datePicker_ValueChanged(object sender, EventArgs e)
        {
            refreshReservations();
        }
        public void tableDoubleClick(object sender, EventArgs e)
        {
            Table table = (Table)sender;
            if (reservationListbox.SelectedItem != null && table.isOpen)
            {
                var selectedItem = reservationListbox.SelectedItem.ToString();
                var name = selectedItem.Split(' ')[7] +" "+ selectedItem.Split(' ')[8];
                DialogResult dialogResult = MessageBox.Show("Are you sure you want to assign "+name+" to Table "+table.tableName.Text+" ?","Sit customer", MessageBoxButtons.YesNo);
                if (dialogResult == DialogResult.Yes)
                {
                    int reservationId = Convert.ToInt32(reservationListbox.SelectedItem.ToString().Split(',')[0].Split('=')[1]);
                    service.deleteReservation(reservationId);
                    this.refreshReservationsDB();
                    table.sitGuest(this,new EventArgs());
                }
            }
        }
    }
}
```

ReservationForm

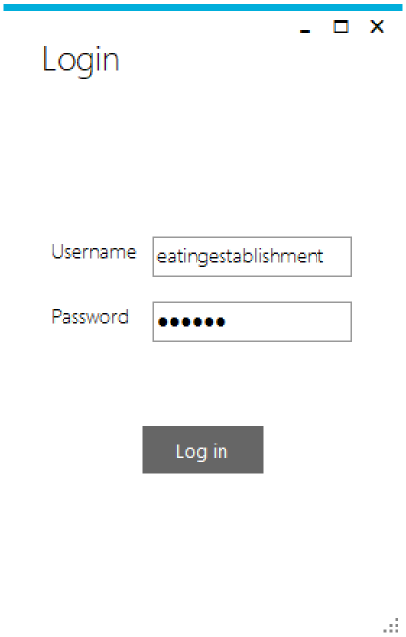
```
 public partial class ReservationForm : MetroFramework.Forms.MetroForm
    {
        User user;
        WebService1SoapClient service;
        Reservation reservation;
        RestaurantForm restaurantForm;
        public ReservationForm(User user, RestaurantForm restaurantForm)
        {
            this.user = user;
            this.service = restaurantForm.service;
            this.restaurantForm = restaurantForm;
            InitializeComponent();
        }
       public ReservationForm(Reservation reservation,User user,RestaurantForm restaurantForm)
        {
            this.user = user;
            this.service = restaurantForm.service;
            this.reservation = reservation;
            this.restaurantForm = restaurantForm;
            InitializeComponent();
            this.Text = "Edit Reservation";
            this.firstNameTextbox.Text = reservation.FirstName;
            this.lastNameTextbox.Text = reservation.LastName;
            this.PhoneTextbox.Text = reservation.PhoneNumber;
            this.PartySizeTextbox.Text = reservation.PartySize.ToString();
            this.reservationDatepicker.Value = reservation.ReservationDate;
            this.reservationHourTextbox.Text = reservation.ReservationTime.Hours.ToString();
            this.reservationMinuteTextbox.Text = reservation.ReservationTime.Minutes.ToString();

            this.descriptionTextbox.Text = reservation.ReservationDetails;
            this.submitButton.Text = "Edit";
            this.deleteButton.Visible = true;
        }
        private void submitButton_Click(object sender, EventArgs e)
        {
            string Firstname = this.firstNameTextbox.Text;
            string Lastname = this.lastNameTextbox.Text;
            string Phonenumber = this.PhoneTextbox.Text;
            int PartySize = Convert.ToInt32(this.PartySizeTextbox.Text);
            int Hour = Convert.ToInt32(this.reservationHourTextbox.Text);
            int Minute = Convert.ToInt32(this.reservationMinuteTextbox.Text);
            DateTime Date = this.reservationDatepicker.Value;
            string Description = this.descriptionTextbox.Text;
            if(submitButton.Text == "Edit")
            {
                this.service.editReservation(this.reservation.ReservationId,this.reservation.ClientId,Firstname, Lastname, Phonenumber, PartySize, Date, Hour, Minute, Description, this.user.Id);
                this.restaurantForm.refreshReservationsDB();
                this.Close();
            }
            else
            {
                this.service.addReservation(Firstname, Lastname, Phonenumber, PartySize, Date, Hour,Minute, Description,this.user.Id);
                this.restaurantForm.refreshReservationsDB();
                this.Close();
            }
        }
        private void deleteButton_Click(object sender, EventArgs e)
        {
            DialogResult dialogResult = MessageBox.Show("Are you sure you want to delete the reservation?", "Delete reservation", MessageBoxButtons.YesNo);
            if (dialogResult == DialogResult.Yes)
            {
                this.service.deleteReservation(this.reservation.ReservationId);
                this.restaurantForm.refreshReservationsDB();
            }
            this.Close();
        }
    }
```

Screenshots:

Pe partea de client am folosit ca librării externe Newtonsoft.Json și MetroFramework.

Login

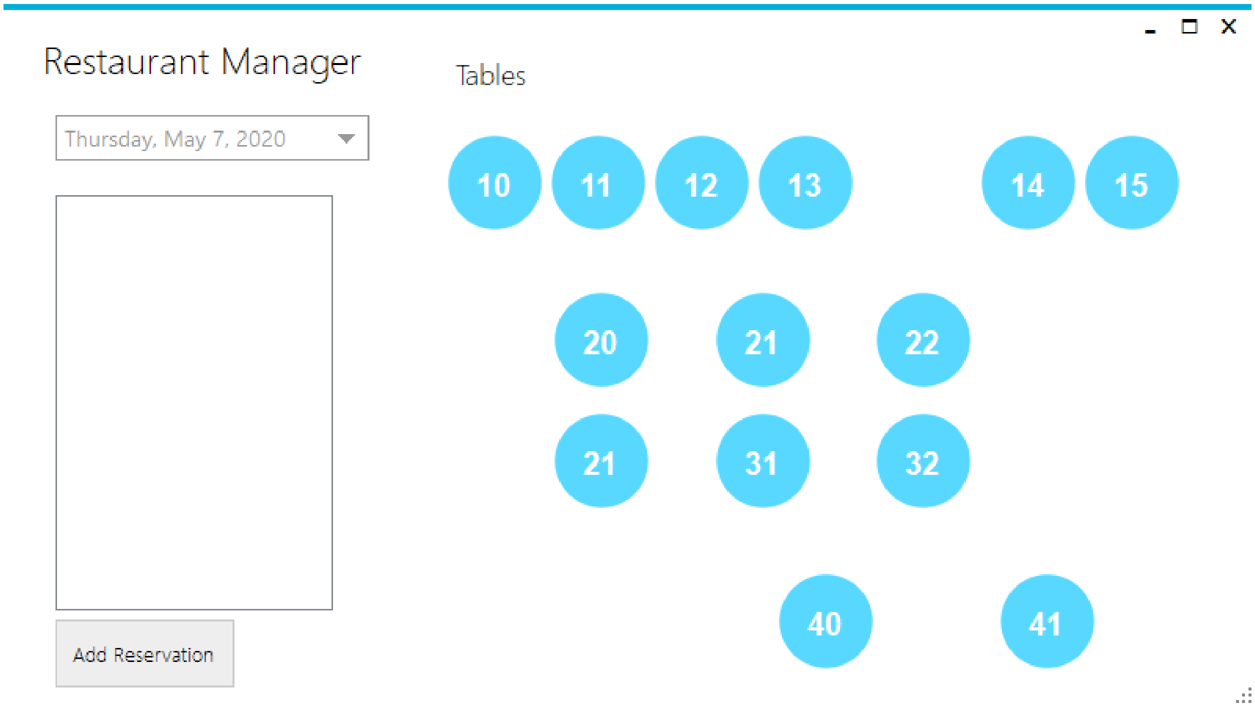


Login

Username eatingestablishment

Password ●●●●●●

Restaurant Form

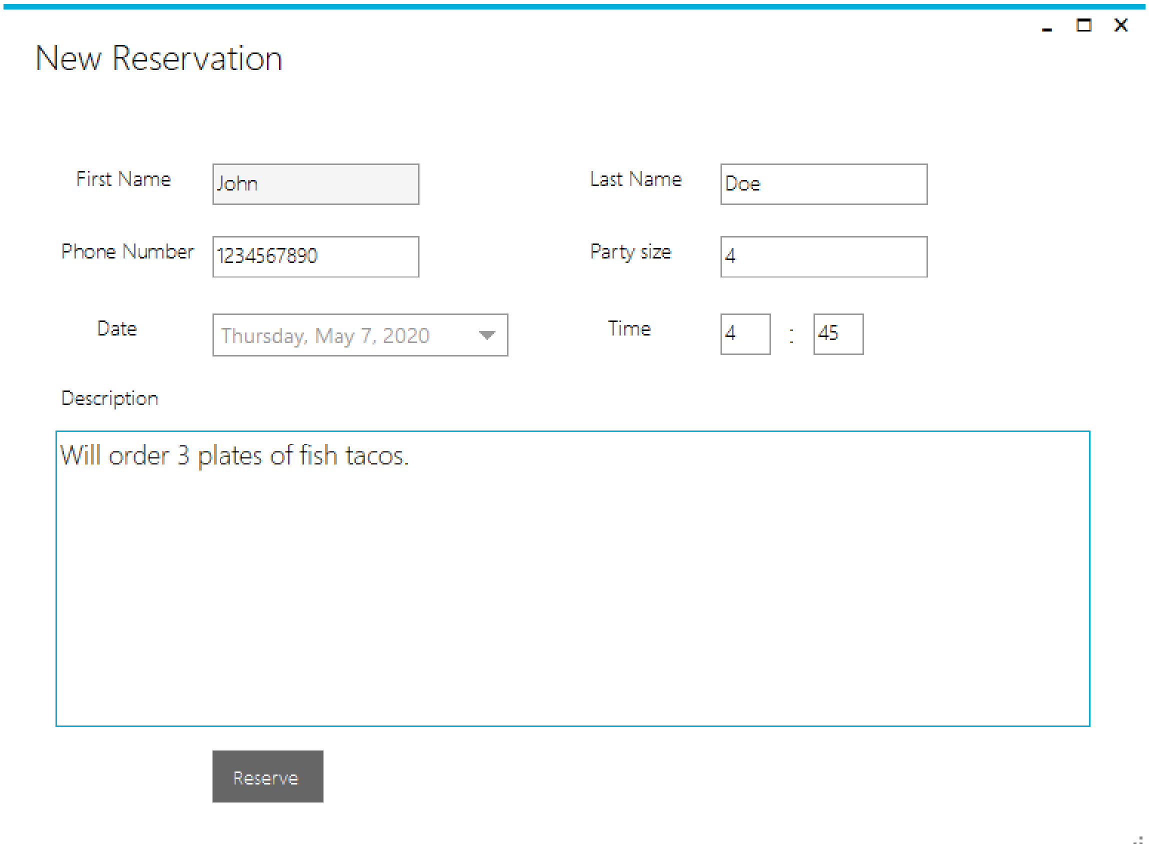


Restaurant Manager

Thursday, May 7, 2020

Rezervation Form

Adăugare rezervare



First Name: John

Last Name: Doe

Phone Number: 1234567890

Party size: 4

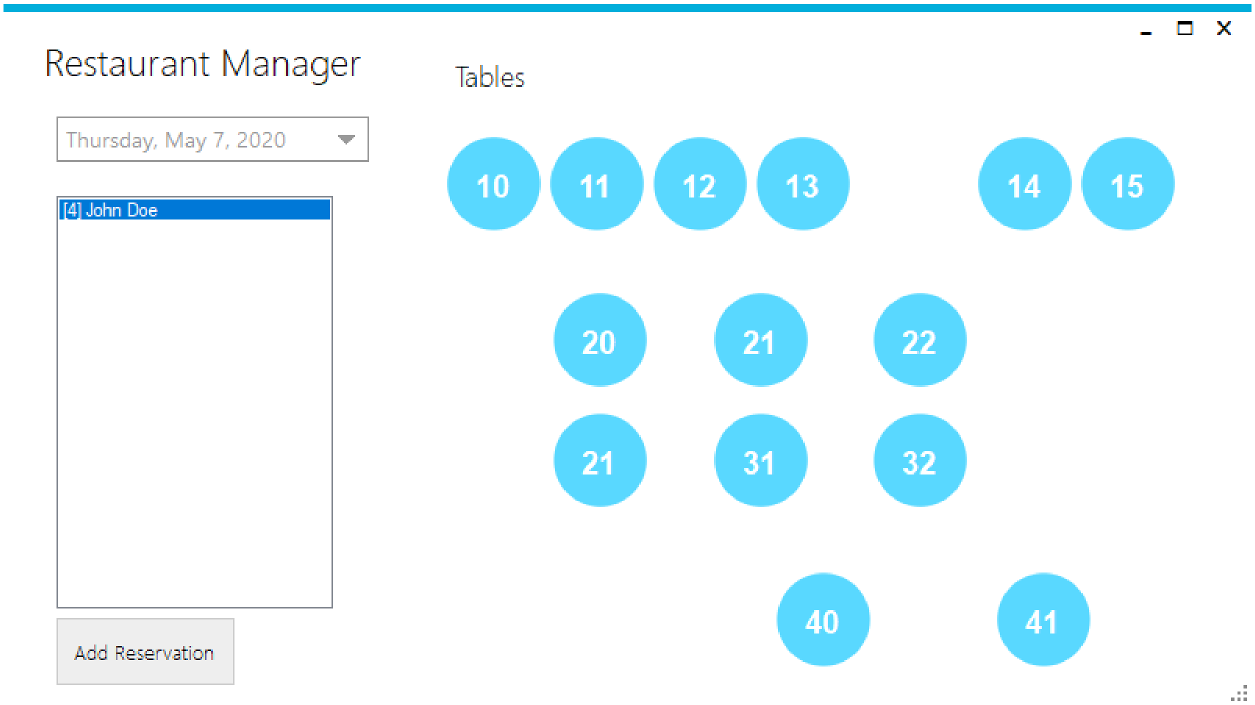
Date: Thursday, May 7, 2020

Time: 4 : 45

Description

Will order 3 plates of fish tacos.

Rezervarea a fost adaugată în baza de date



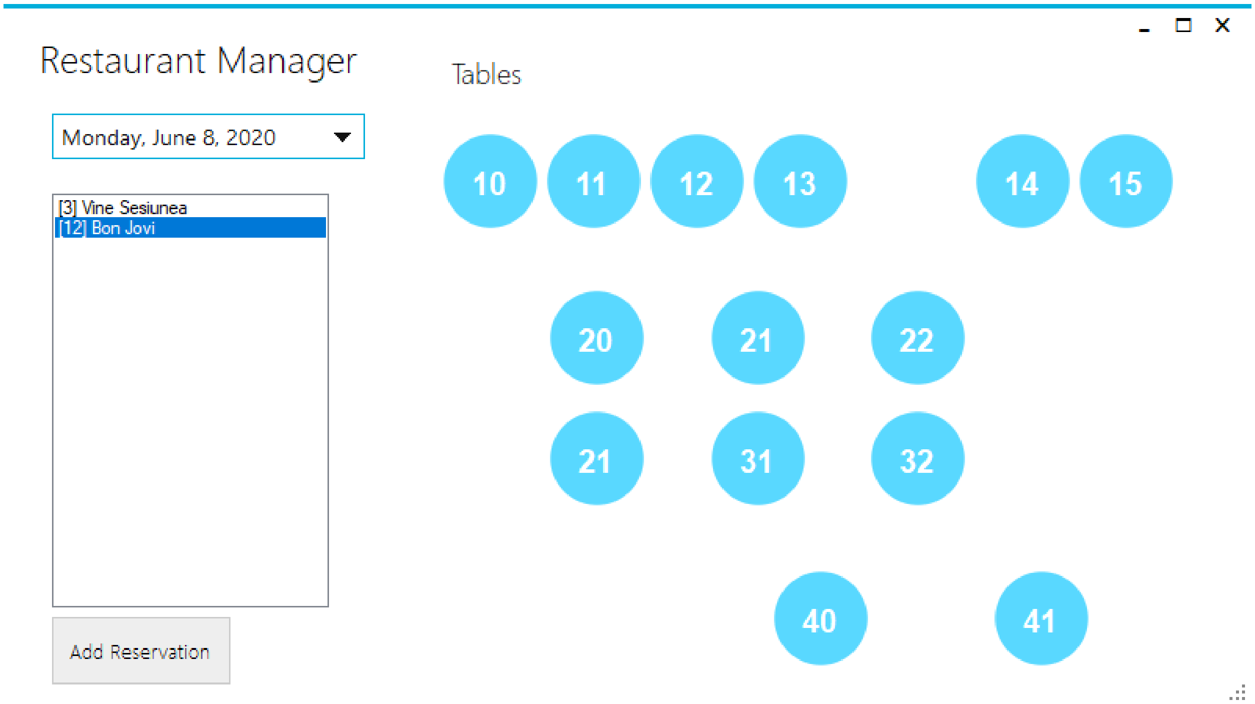
Thursday, May 7, 2020

[4] John Doe

Utilizatorul a fost și el adăugat dacă nu exista cumva deja, caz în care i s-a asociat doar o nouă rezervare.

Filtrare rezervări pe dată

Am schimbat data pentru a filtra reservările.

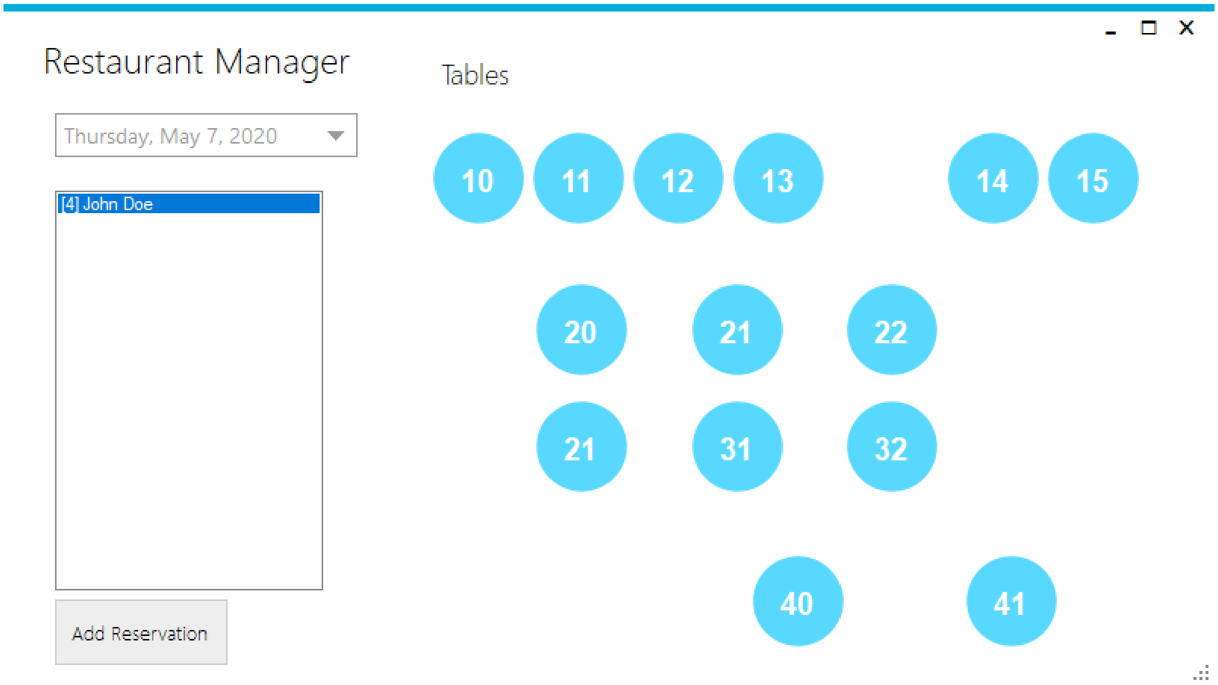


Restaurant Manager

Monday, June 8, 2020

[3] Vine Sesiunea
[12] Bon Jovi

Editare rezervare

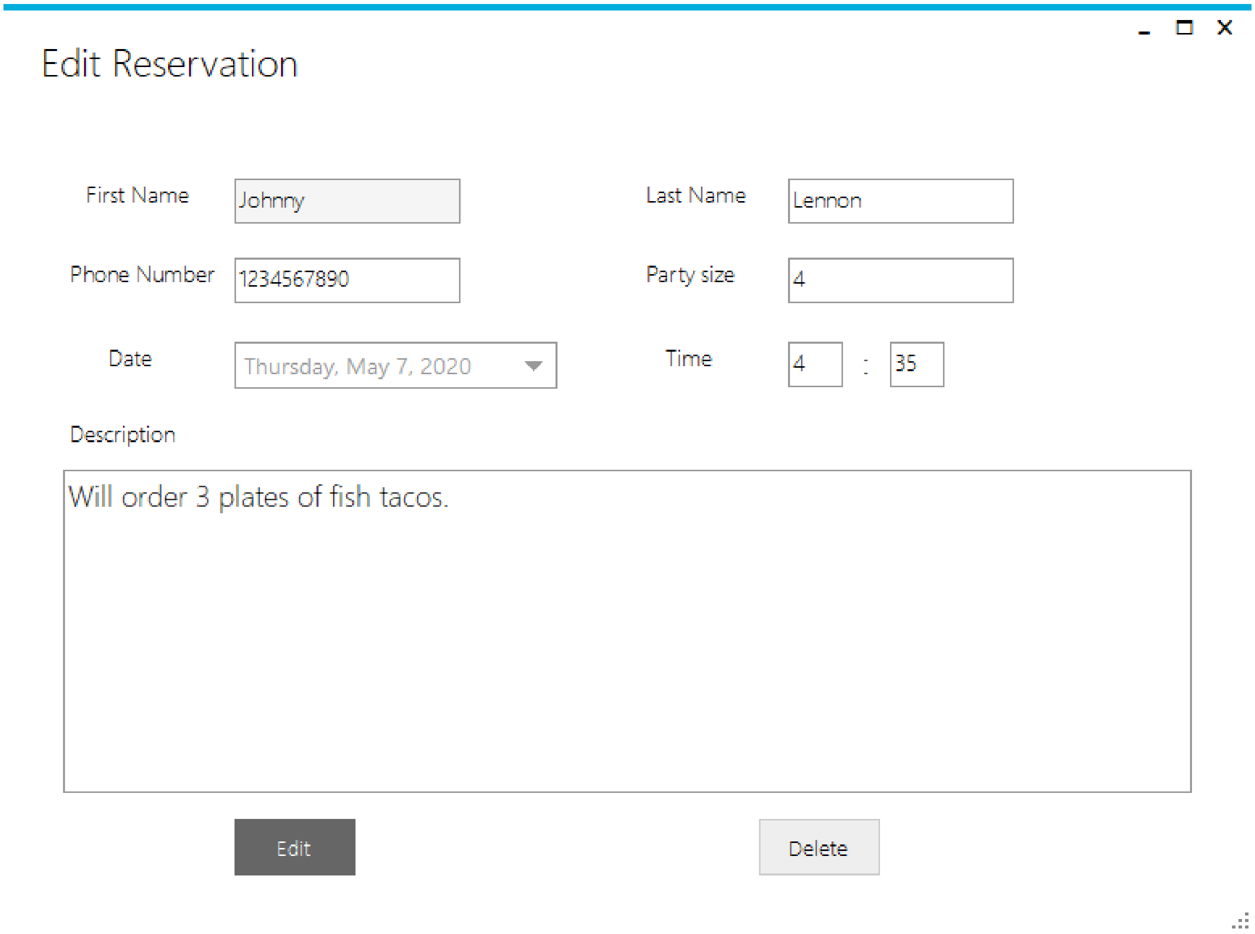


Restaurant Manager

Thursday, May 7, 2020

[4] John Doe

Dacă facem dublu click pe o rezervare se va deschide din nou Rezervation Form dar de data asta in modul de editare/stergere.



Edit Reservation

First Name: Johnny

Last Name: Lennon

Phone Number: 1234567890

Party size: 4

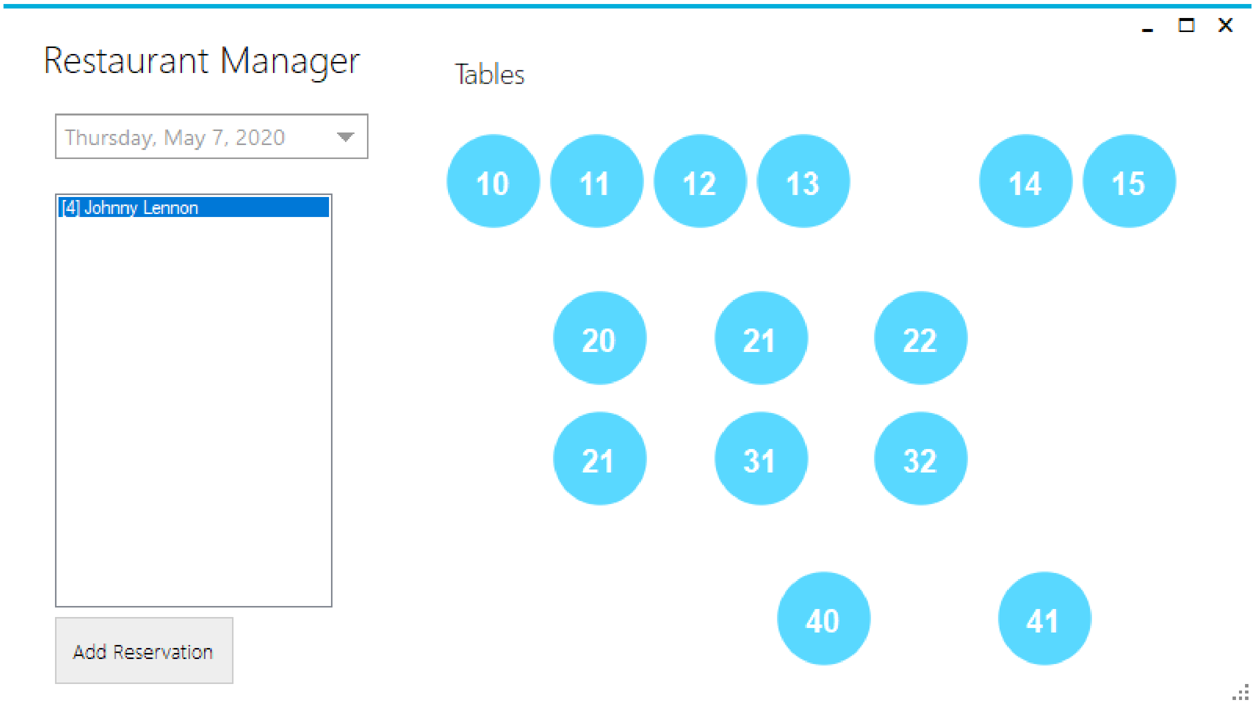
Date: Thursday, May 7, 2020

Time: 4 : 35

Description

Will order 3 plates of fish tacos.

Rezervarea a fost editată

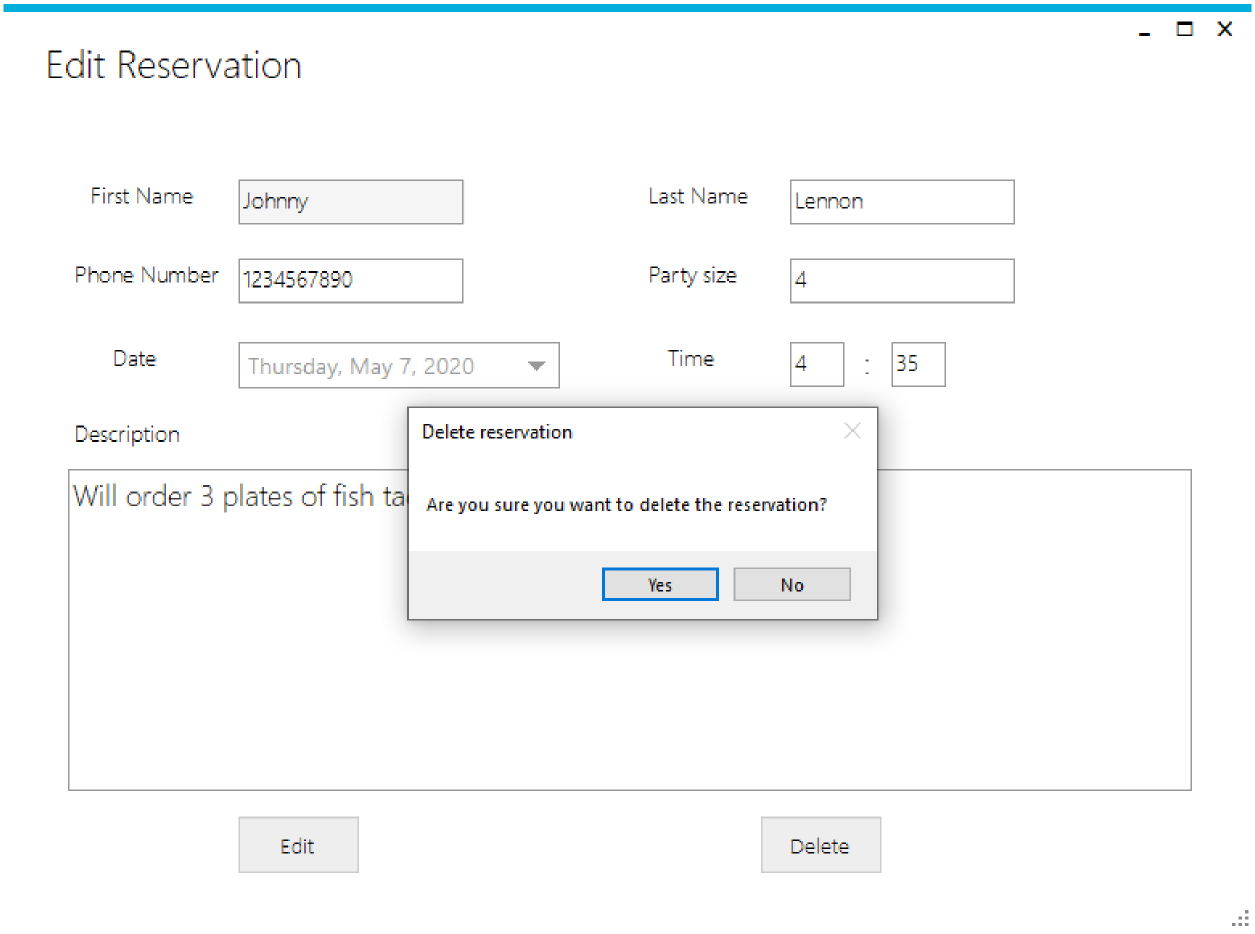


Restaurant Manager

Thursday, May 7, 2020

[4] Johnny Lennon

Ștergere rezervare



Edit Reservation

First Name: Johnny

Last Name: Lennon

Phone Number: 1234567890

Party size: 4

Date: Thursday, May 7, 2020

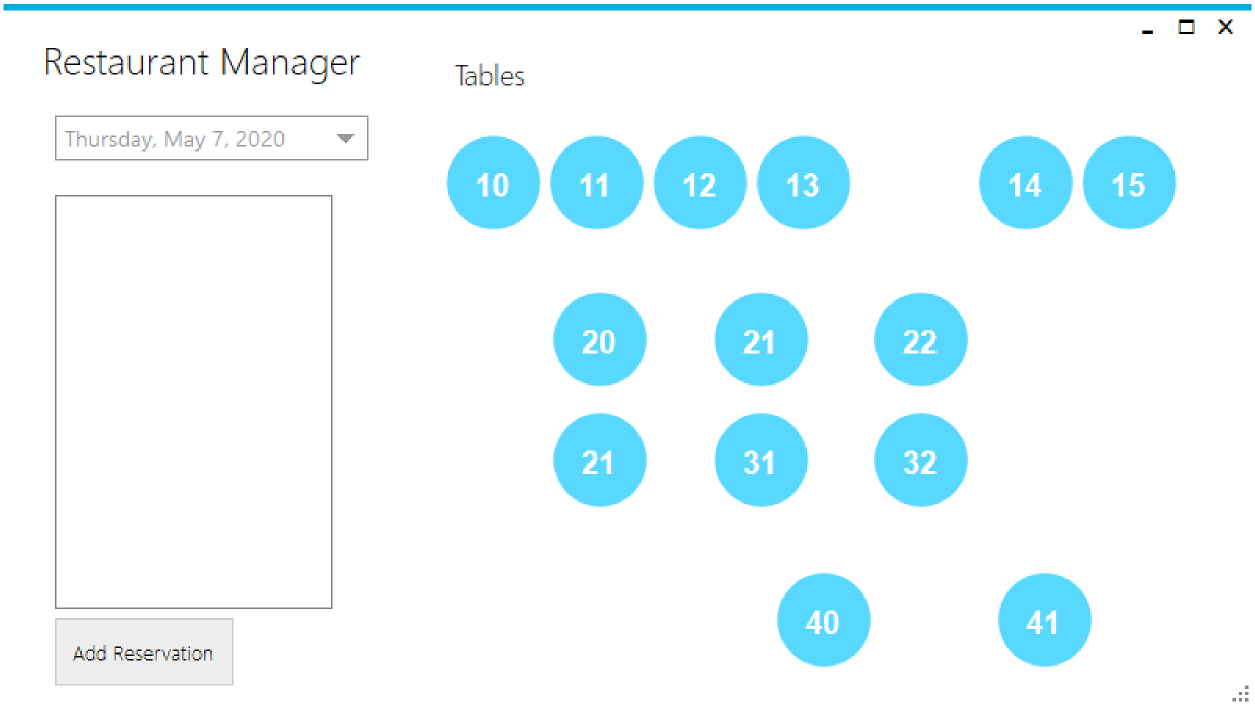
Time: 4 : 35

Description: Will order 3 plates of fish ta

Delete reservation

Are you sure you want to delete the reservation?

Rezervarea a fost ștearsă



Restaurant Manager

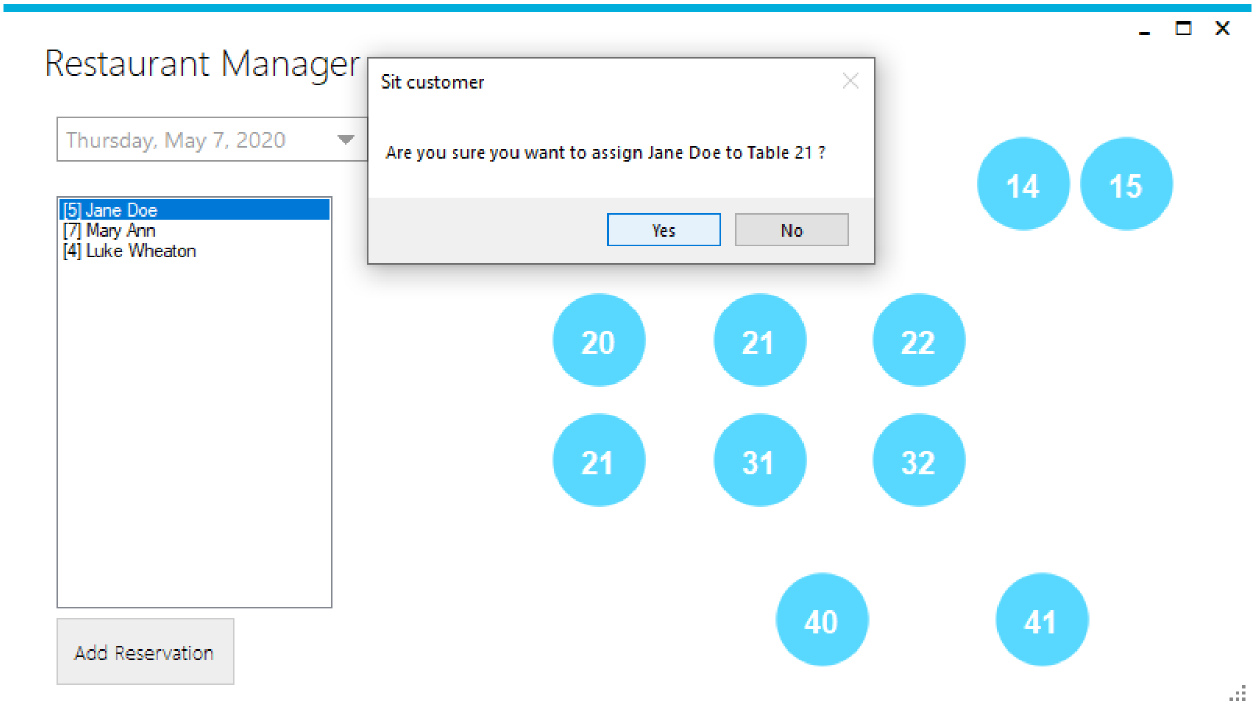
Thursday, May 7, 2020

Până acum am arătat funcționalitățile de Create Read Update Delete prin intermediul care se realizează prin intermediul Serviciului Web, în continuare vor fi prezentate functionalitățile meselor.

Seat clients

Odată ce clienții ce și-au făcut rezervare apar restaurant, ar fi utilă aceasta functionalitate în situatiile in care restaurantul este plin sau are foarte multe rezervari. Putem așeza clienții la o anumită masă și să o marcăm ca fiind ocupată.

Dăm dublu click pe Jane Doe , după care vom da click pe o masă oarecare.



Restaurant Manager

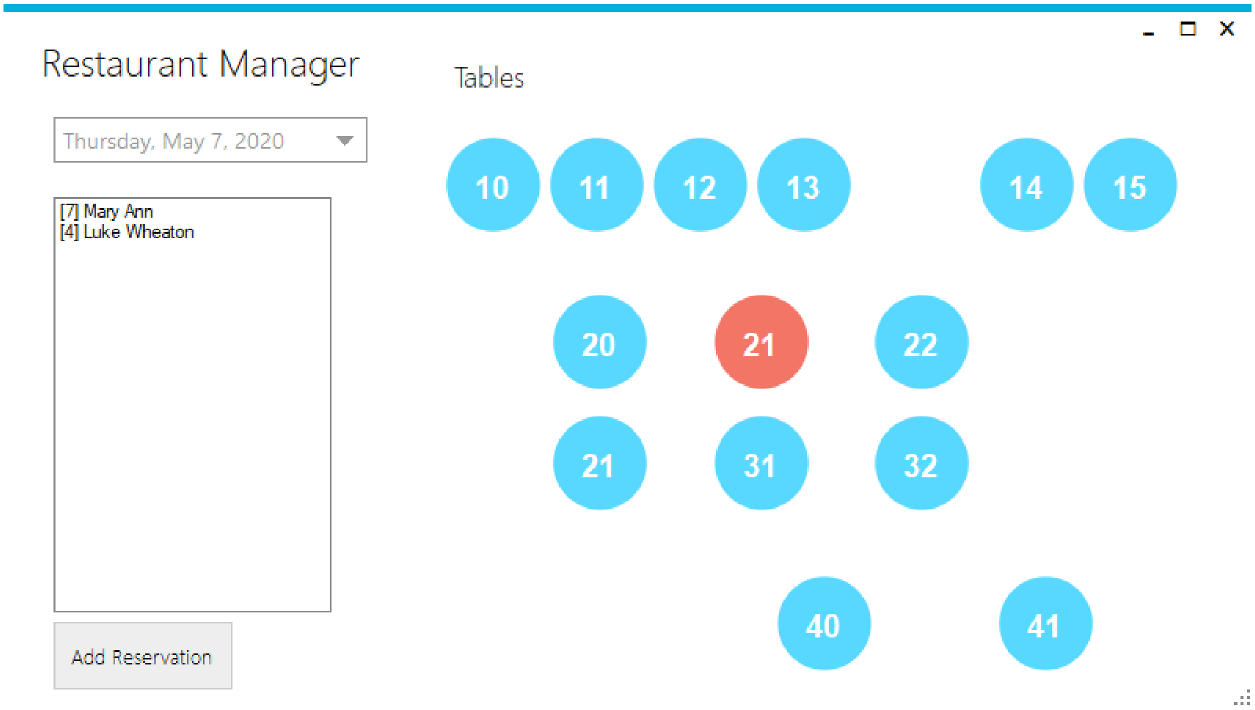
Thursday, May 7, 2020

[5] Jane Doe
[7] Mary Ann
[4] Luke Wheaton

Sit customer

Are you sure you want to assign Jane Doe to Table 21 ?

Clienta Jane Doe a fost eliminată de la Rezervări deoarece am așezat-o la masa 21.



Restaurant Manager

Tables

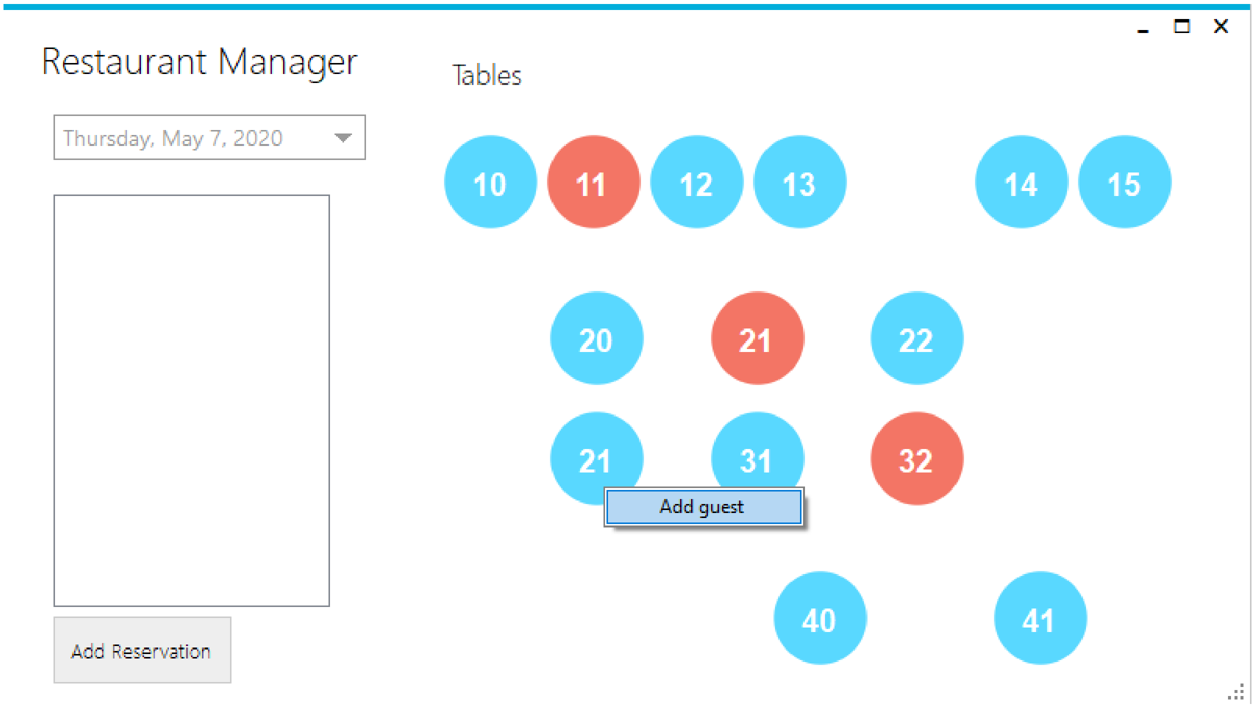
Thursday, May 7, 2020

[7] Mary Ann
[4] Luke Wheaton

Același lucru putem face si cu restul clienților.

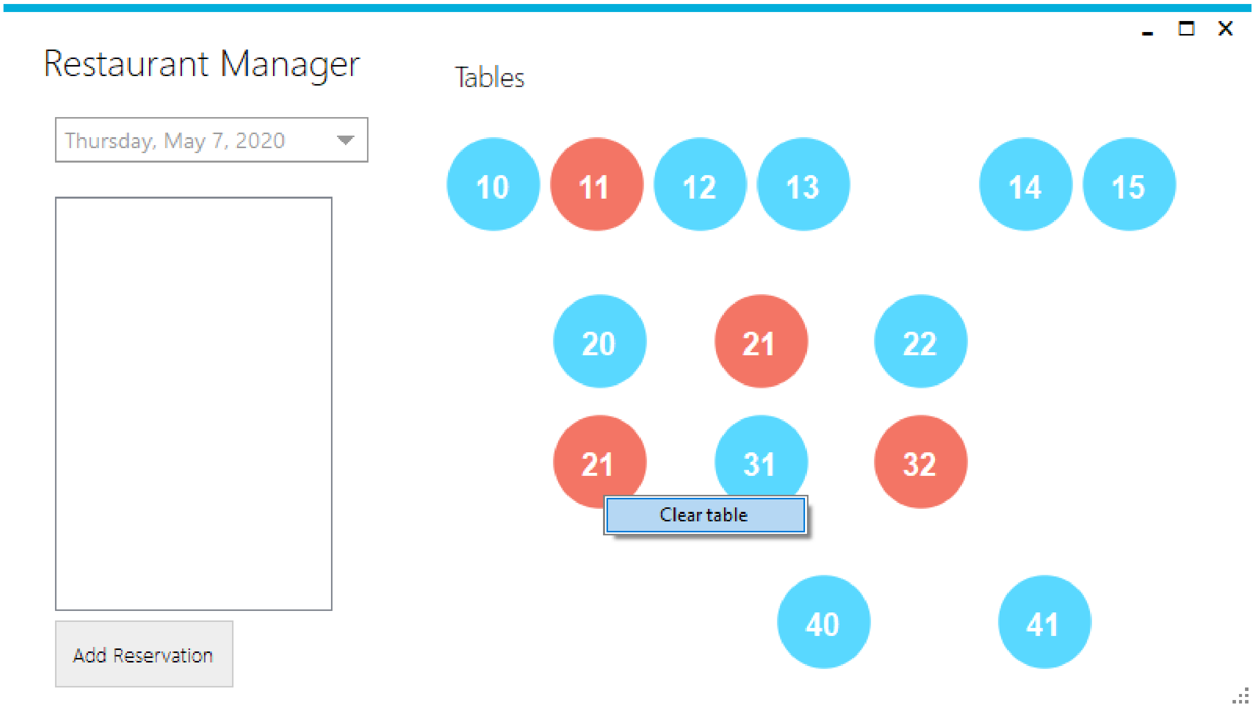
Seat Guest

Există posibilitate ca să apară clienți care nu și-au făcut rezervare. În acest caz se va da click dreapta pe o masă liberă și se va alege “Add Guest”.

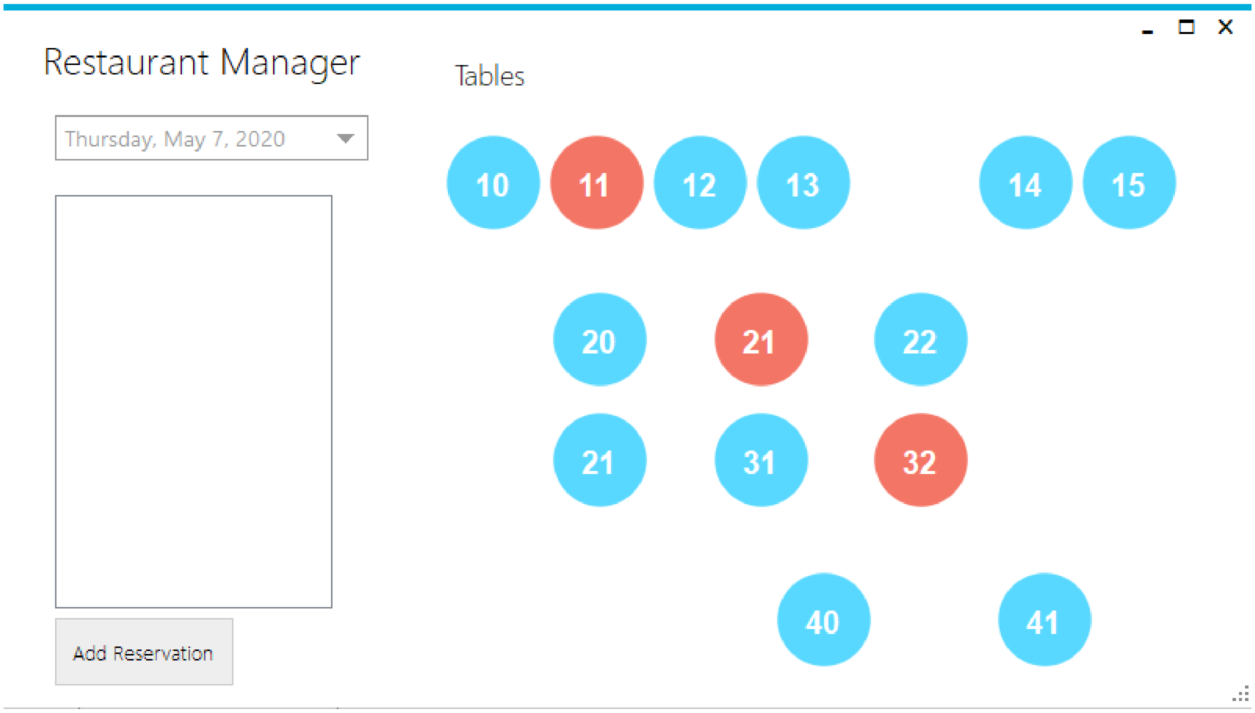


Clear Table

În momentul în care o clienții se vor ridica de la o masă, va exista posibilitatea de a redeschide masa pentru următorii clienți, cu click dreapta “Clear table”.



Masa poate fi din nou ocupată de alți clienți.


Restaurant Manager
Tables
Thursday, May 7, 2020
10
11
12
13
14
15
20
21
22
21
31
32
40
41
Add Reservation