

מטלה 4 שאלה 2

ממשק קניית כרטיסים ליום הסטודנט

מה בממשק?

1. הלקוח יגיע למסך בו ימלא את פרטיו האישיים, לאחר שסימן האם חבר אגודה או לא ייפתח בפניו אפשרות בחירה לכרטיס למסיבה ולכרטיס לאוטובוס, ובאפשרותו לבחור אחד מהם או את שניהם בהתאם לצורך שלו. על מנת להמשיך לחלק הבא, על המשתמש ללחוץ על כפתור "המשך".
2. ייפתח החלק השני במסך בו הלקוח יקבל אינדיקציה של תמחור הכרטיסים שסימן בהתאמה לבחירתו בצג "חבר אגודה". (סטודנט שהינו חבר אגודה מקבל הנחה עבור כל אחד מסוגי הכרטיסים, ואילו סטודנט שאינו חבר ישלם מחיר מלא).
3. כעת, באותו אזור בממשק, כדי להשלים את הרכישה על המשתמש ללחוץ על כפתור התשלום, שגם הוא יהיה בהתאמה לסטטוס "חבר אגודה".

חבר אגודה – תשלום אוטומטי מפרטי כרטיס האשראי השמורים תחת המשתמש שלו באתר האגודה (כביכול מערכת הממשק מזהה את פרטי השם ותעודת הזהות שלו השמורים תחת המשתמש שלו בחשבון האגודה וכך מבצעת שיוך ישיר). תופיע הודעה למשתמש לאחר הלחיצה על הכפתור כמובן שתיידע אותו שזהו תהליך התשלום עבורו.

אינו חבר אגודה – ייפתח בפני הקונה אפשרות לבחירת אופן התשלום:

- תשלום דרך "ביט" – בלחיצה על כפתור זה, ייפתח האתר הרשמי של ביט לצורך העברת התשלום (בפועל נפתח האתר של ביט אך באופן התיאורטי, הקישור יהיה ישירות למסך תשלום המקושר לחשבון מארגן האירוע ועם פרטי התשלום שתומחרו לו מבחירתו בהזמנה- דבר הקיים היום באפליקציות מסוג זה).
- תשלום במקום האירוע – בלחיצה על כפתור זה תופיע הודעה למשתמש אשר תיידע אותו שעל אחריותו להביא עימו את אמצעי התשלום (מזומן/אשראי) על מנת לשלם בכניסה, אחרת כרטיסו לא יהיה בתוקף וכניסתו לא תאושר.

4. בשלב הבא, על מנת לבצע את ההזמנה באופן סופי ולשלוח למערכת את פרטי ההזמנה והתשלום, על המשתמש ללחוץ על כפתור "סיום הזמנה" (המשתמש מקבל הודעה סופית לאחר שלב התשלום בה הוא מיודע שכדי לסיים את התהליך עליו ללחוץ על כפתור זה).

5. לאחר לחיצה על כפתור "סיום הזמנה", ייפתח בפני המשתמש חלונית חדשה שהיא מסך סיכום ההזמנה שלו. בחלונית זו המשתמש יראה את פרטי ההזמנה: שעת ביצוע, מספר כרטיסים בהזמנה, סכום מצטבר עבור הקנייה, פרטיו האישיים כפי שמילא, סוגי הכרטיסים שרכש והמחיר עבור כל אחד מהם, ואופן התשלום שבחר במידה ואינו חבר אגודה.

סביבת הממשק:

לצד חלוניות פרטי הקנייה, מופיע לוגו של אגודת רופין, מופיע הזמן (בשניות) הנותן אינדיקציה למשתמש כמה זמן עבר מאז פתיחת החלונית ותחילת תהליך הרכישה (מימוש של הפקד timer), ובנוסף מופיע צג בסגנון "בלוק טעינה" הנותן אינדיקציה ויזואלית למשתמש על שלב הימצאותו בתהליך הרכישה ועד כמה הוא "מתקרב" לסיומו.

הפקדים והאירועים שבממשק:

- Text Box – שדות מילוי פרטי שם פרטי, שם משפחה, ת.ז. בנוסף שדות שמודפסים להם פרטים (מחיר למשל).
- Label – כותרות, כותרות לשדות, כותרת להצגת שעת ביצוע ההזמנה (בטופס השני).
- Progress Bar – פקד לאינדיקציה המתמלא על פי התקדמות בשלבי התהליך.
- Timer – סופר את השניות שעברו מעת פתיחת הטופס ועד לחיצה על כפתור סיום הזמנה.
- Combo Box – בחירת תשובה יחידה ל "האם אתה חבר אגודה?" מתוך 2 אפשרויות.
- Check Box – בחירת תשובה (לא בהכרח יחידה!) לסוג הכרטיס.
- Button – מספר כפתורים עם התיאור הרלוונטי לשלב בו נמצאים ואירועים בהתאמה:
 - * ישנם כפתורים שכלל אינם מופיעים על המסך עד שלא מתבצע השלב הקודם שהוא כהתניה ללחיצה על אותו הכפתור, ישנם כפתורים שנראים אך לא ניתנים ללחיצה מאותה סיבה.
 - * על פי תפקיד הכפתור בממשק, במידה ונלחץ, מופיעות הודעות בהתאם לנושא הכפתור המודיעות ללקוח שלחיצתו התקבלה. במידה ונלחץ כפתור לפני מילוי או בחירת הנתונים הנדרשים למעבר לשלב הבא- גם במקרה זה תופיע הודעה עם מלל מתאים לבעיה.
 - * לחיצה על כפתור "BIT" גורמת לדפדפן להיפתח ישירות לפי הקישור שבחרנו.
 - * אירוע לחיצה על כפתור "סיום הזמנה" פותח את הטופס השני שבפרויקט, המציג את חלונת פרטי ההזמנה.
- List Box – הצגת פרטי ההזמנה בפורמט של רשימה.

המחלקות הקיימות ותיאורן:

- **מחלקת Client:** שומרת פרטים על הלקוח המבצע את ההזמנה: שם פרטי, שם משפחה, ת.ז, האם חבר אגודה, אופן תשלום.
 - הבנאי שלה מקבל ארגומנטים בהתאמה לשדות הנ"ל, ואיתחולם מתבצע ע"י setters הנמצאים ב-properties. setters השתמשו במנגנון ה- throw exceptions וכתבנו הודעות בהתאמה לשגיאה, המתקבלות בטופס Form1 על מנת ליידע את המשתמש בכך.
- **מחלקת Ticket:** שומרת את סוג הכרטיס שנבחר, מחיר סופי לכרטיס, ואת הלקוח שעל שמו הכרטיס. בנוסף מכילה שדות עם ערכים קבועים עבור מחירי הכרטיסים ההתחלתיים וגובה ההנחה עבור חבר אגודה.
 - בנאי המחלקה מקבל אובייקט מסוג Client על מנת לשייך את פרטי הלקוח לכרטיס הספציפי וגם כדי לקבוע (ולאתחל) עפ"י תכונת חבר אגודה את המחיר הסופי. בנוסף מקבל ארגומנט לאתחול סוג הכרטיס לפי מה שסימן הלקוח בממשק. המחלקה מכילה מתודות set ו-get לכל השדות.
- **מחלקת TicketsList:** זוהי מחלקה סטטית, מתפקדת כמחלקת שירות עבור הוספת כרטיס למערך ההזמנה. על כן היא מכילה שדה סטטי שסופר את מספר הכרטיסים שבהזמנה, ומתודת AddTicket.
- עבור כל סוג כרטיס, ובפרט לגבי אופן התשלום שנבחר על ידי לקוח שאינו חבר אגודה, נשמרים פרטיו האישיים ובחירת התשלום, עבור כל אחד מסוגי הכרטיסים. ראינו חשיבות בשמירת הפרטים עבור כל כרטיס בנפרד על מנת שכאשר אותו לקוח יגיע להסעה או למסיבה, העובד שבודק את פרטיו באותה מערכת יראה שאכן הכרטיס שבידו הוא תחת שמו ויוכל לאשר את כניסתו.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.Linq;
using System.Linq.Expressions;
using System.Reflection;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Net.WebRequestMethods;

namespace StudentDay
{
    public partial class Form1 : Form
    {
        int ticks = 0;
        Ticket[] ticketsList = new Ticket[0];
        public bool member;
        const string URL = "https://www.bitpay.co.il/he";

        public Form1()
        {
            InitializeComponent();
            timer1.Start();
            this.Text = "אגודת הסטודנטים רופין- מסך הזמנה";

            progressBar1.Maximum = 5;
            progressBar1.Minimum = 0;
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            MessageBox.Show("ARE YOU READY TO CELEBRATE ??");
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            ticks++;
            lblTime.Text = ticks.ToString();
        }

        private void splitContainer1_Panel1_Paint(object sender, PaintEventArgs e)
        {
        }

        private void cmbIsMember_SelectedIndexChanged(object sender, EventArgs e)
        {
            progressBar1.Value = 1;
            if (cmbIsMember.SelectedIndex == 1 ||
                cmbIsMember.SelectedIndex == 2)
            {
                lblTicketToBuy.Visible = true;
                chkBoxParty.Visible = true;
                chkBoxBus.Visible = true;
            }
        }
    }
}
```

```

        if (cmbIsMember.SelectedIndex == 0)
        {
            MessageBox.Show("Please select your member option", "שגיאה  
במילוי הפרטים", MessageBoxButtons.OK, MessageBoxIcon.Question);
            lblTicketToBuy.Visible = false;
            chkBoxParty.Visible = false;
            chkBoxBus.Visible = false;
        }
    }

e) private void chkBoxParty_CheckedChanged(object sender, EventArgs
    {
        if (chkBoxParty.Checked)
        { btnContinue.Enabled = true; }
    }

    private void chkBoxBus_CheckedChanged(object sender, EventArgs e)
    {
        if (chkBoxBus.Checked)
        { btnContinue.Enabled = true; }
    }

    private void btnContinue_Click(object sender, EventArgs e)
    {
        progressBar1.Value = 3;

        bool member = false; //default
        if (cmbIsMember.SelectedIndex == 1)
        {
            member = true;
            btnAcceptOrder.Visible = true;
        }
        if (cmbIsMember.SelectedIndex == 2)
        {
            member = false;
            btnPaymentMethod.Visible = true;
        }

        try
        {
            Client client = new Client(txtFirstName.Text,
            txtLastName.Text, txtID.Text, member); // adding new client

            if (chkBoxParty.Checked) // adding party ticket
            {
                Ticket ticketP = new Ticket(client, "מסיבה");
                ticketsList = TicketsList.AddTicket(ticketP,
            ticketsList);

                lblPartyTickPrice.Visible = true;
                txtPartyTickPrice.Visible = true;
                foreach (var t in ticketsList)
                {
                    if (t.GetTypeOfTicket() == "מסיבה" &&
            (t.Client.ID == txtID.Text))
                    {
                        txtPartyTickPrice.Text =
            (t.GetPrice()).ToString();
                    }
                }
            }
            if (chkBoxBus.Checked) // adding bus ticket
            {
                Ticket ticketB = new Ticket(client, "אוטובוס");
            }
        }
    }

```

```

ticketsList);
        ticketsList = TicketsList.AddTicket(ticketB,
        lblBusTickPrice.Visible = true;
        txtBusTickPrice.Visible = true;

        foreach (var t in ticketsList)
        {
            if (t.GetTypeOfTicket() == "אוטובוס" &&
(t.Client.ID == txtID.Text))
            {
                txtBusTickPrice.Text =
(t.GetPrice()).ToString();
            }
        }
    }
    catch (Exception exp)
    {
        MessageBox.Show(exp.Message);
    }
}
private void lblPartyTickPrice_Click(object sender, EventArgs e)
{
}

private void txtPartyTickPrice_TextChanged(object sender,
EventArgs e)
{
}

private void txtBusTickPrice_TextChanged(object sender, EventArgs
e)
{
}

private void btnAcceptOrder_Click(object sender, EventArgs e)
{
    MessageBox.Show("התשלום על הרכישה יבוצע מפרטי כרטיס
לדיעתך, האשראי ששמורים בחשבונך באתר האגודה
לסיום הזמנה לחץ על כפתור סיום יצ"ח. ", "ההזמנה", MessageBoxButtons.OK, MessageBoxIcon.Information);
    btnFinishOrder.Visible = true;
    progressBar1.Value = 5;
}

private void btnPaymentMethod_Click(object sender, EventArgs e)
{
    btnBit.Visible = true;
    btnPayInPlace.Visible = true;
    progressBar1.Value = 4;
}

private void btnBit_Click(object sender, EventArgs e)
{
    Process.Start(URL); // open Bit website with the
organization's payment details
    foreach (var t in ticketsList)
    {
        if (t.Client.ID == txtID.Text)
        {
            t.Client.PaymentMethod = 1;

```

```

    }
}
MessageBox.Show("לסיום הזמנה לחץ על יצא\!\בחירתך התקבלה בהצלחה"
    לסיים הזמנה לחץ על יצא\!\בחירתך התקבלה בהצלחה"
    , "נפתור סיום ההזמנה", MessageBoxButtons.OK);
btnFinishOrder.Visible = true;
progressBar1.Value = 5;
}

private void btnPayInPlace_Click(object sender, EventArgs e)
{
    DialogResult res = MessageBox.Show("שים לב, כניסתך לאירוע תהיה מותנית בביצוע התשלום בכניסה, ניתן במזומן או באשראי לבחירתך.",
    "בחירת אופן תשלום: תשלום במקום", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning);
    if (res == DialogResult.OK)
    {
        MessageBox.Show("לסיום הזמנה יצא\!\בחירתך התקבלה בהצלחה"
            , "לחץ על כפתור סיום ההזמנה",
            MessageBoxButtons.OK);
    }

    foreach (var t in ticketsList)
    {
        if (t.Client.ID == txtID.Text)
        {
            t.Client.PaymentMethod = 2;
        }
    }
    btnFinishOrder.Visible = true;
    progressBar1.Value = 5;
}

private void btnFinishOrder_Click(object sender, EventArgs e)
{
    string NumOfTicket = TicketsList.numOfTickets.ToString();
    timer1.Stop();
    Form2 frm2 = new Form2(ticketsList, NumOfTicket);
    frm2.ShowDialog();
}

private void txtLastName_TextChanged(object sender, EventArgs e)
{
}

private void lblTime_Click(object sender, EventArgs e)
{
}
}
}

```

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace StudentDay
{
    public partial class Form2 : Form
    {
        Ticket[] TicketsList;
        public Form2(object[] arrOfTickets, string numOfTicket)
        {
            InitializeComponent();
            this.Text = "אגודת הסטודנטים רופ"ן - מסך סיכום הזמנה";

            TicketsList = (Ticket[])arrOfTickets;

            // printing details:

            lstOrderDetails.Items.Add("שם מלא:" + " " +
TicketsList[0].Client.FirstName.ToString() + " " +
TicketsList[0].Client.LastName.ToString());
            lstOrderDetails.Items.Add("תעודת זהות:" +
TicketsList[0].Client.ID.ToString());

            foreach (Ticket ticket in TicketsList)
            {
                lstOrderDetails.Items.Add("סוג כרטיס:" +
ticket.GetTypeOfTicket().ToString());
                lstOrderDetails.Items.Add("מחיר:" +
ticket.GetPrice().ToString());

                if (ticket.Client.PaymentMethod == 1)
                {
                    lstOrderDetails.Items.Add("תשלום דרך אתר בית");
                }
                if (ticket.Client.PaymentMethod == 2)
                {
                    lstOrderDetails.Items.Add("תשלום במקום האירוע במזומן/באשראי");
                }
            }

            double dTotalPrice = 0;
            foreach (Ticket ticket in TicketsList)
            {
                dTotalPrice += ticket.GetPrice();
            }
            lblTotPriceForPrint.Text = dTotalPrice.ToString();

            lblTicketsNumForPrint.Text = numOfTicket;
        }
    }
}
```

```

        lblTimeForPrint.Text = DateTime.Now.ToLongTimeString();
    }

    private void Form2_Load(object sender, EventArgs e)
    {

    }

    private void lblTitle_Click(object sender, EventArgs e)
    {

    }

    private void lstOrderDetails_SelectedIndexChanged(object
sender, EventArgs e)
    {

    }

    private void lblTicketsNumForPrint_Click(object sender,
EventArgs e)
    {

    }
}
}

```

:מחלקת Program

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace StudentDay
{
    internal static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

```



```
using System;
using System.Collections.Generic;
using System.Drawing.Imaging;
using System.Linq;
using System.Reflection;
using System.Text;
using System.Threading.Tasks;

namespace StudentDay
{
    internal class Client
    {
        string firstName;
        string lastName;
        string id;
        bool isMember;
        int paymentMethod;

        public Client(string FirstName, string LastName, string ID, bool
IsMember) // constructor
        {
            this.FirstName = FirstName;
            this.LastName= LastName;
            this.ID= ID;
            this.IsMember = IsMember;
            paymentMethod = 0;
        }
        public string FirstName
        {
            get { return firstName; }
            set // checking if first name is correct
            {
                if (value == null || value == "")
                {
                    throw new Exception("No value was entered in the
required field");
                }

                firstName = value;
            }
        }
        public string LastName
        {
            get { return lastName; }
            set // checking if last name is correct
            {
                if (value == null || value == "")
                {
                    throw new Exception("No value was entered in the
required field");
                }

                lastName = value;
            }
        }
        public string ID
        {
            get { return id; }
            set // checking if id is correct - with 9 digit
            {
```

```

        if (value == null)
        {
            throw new Exception("No value was entered in the
required field");
        }

        if (value.Length != 9)
        {
            throw new Exception("ID must have exactly 9 digits.");
        }
        id = value;
    }
}

public bool IsMember
{
    get { return isMember; }
    set // checking if membership exist
    {
        isMember = value;
    }
}

public int PaymentMethod
{
    get { return paymentMethod; }
    set
    {
        if (!this.isMember)
        {
            paymentMethod = value; // 1-Bit, 2-PayInPlace, in
adjust to the clicked button
        }
    }
}
}
}
}
}

```

```
using System;
using System.Collections.Generic;
using System.Drawing.Imaging;
using System.Linq;
using System.Security.Policy;
using System.Text;
using System.Threading.Tasks;

namespace StudentDay
{
    internal class Ticket
    {
        public const int PARTY_TIC_PRICE = 80;
        public const int BUS_TIC_PRICE = 10;
        const double DISCOUNT_MEMBER = 0.2;

        string typeOfTicket;
        double price;
        Client client;

        public Ticket(Client client, string type) // c'tor
        {
            this.client = client;
            SetType(type);
            SetPrice(); //client
        }

        public void SetType(string type)
        {
            // NO NEED: if (type == null || type == "") because in the
            Form1 its not possible to arrive the c'tor without choosing a type from
            the options

            if (type == "מסיבה")
            {
                typeOfTicket = type;
            }

            if (type == "אוטובוס")
            {
                typeOfTicket = type;
            }
        }

        public string GetTypeOfTicket() { return typeOfTicket; }

        public void SetPrice ()
        {
            if (this.client.IsMember==true)
            {
                double FinalPrice = 0;
                if (typeOfTicket == "מסיבה")
                {
                    FinalPrice = PARTY_TIC_PRICE * (1-DISCOUNT_MEMBER);
                }

                if (typeOfTicket == "אוטובוס")
                {
                    FinalPrice = BUS_TIC_PRICE * (1-DISCOUNT_MEMBER);
                }
            }
        }
    }
}
```

```

        price = FinalPrice;
    }
    else
    {
        if (typeOfTicket == "מסיבה") { price = PARTY_TIC_PRICE; }
        if (typeOfTicket == "אוטובוס") { price = BUS_TIC_PRICE; }
    }
}

public double GetPrice() { return price; }

public Client Client
{
    get { return client; }
    set { client = value; } // the Client class already have
setters for itself so its sure its set
}
}

```

מחלקת TicketsList:

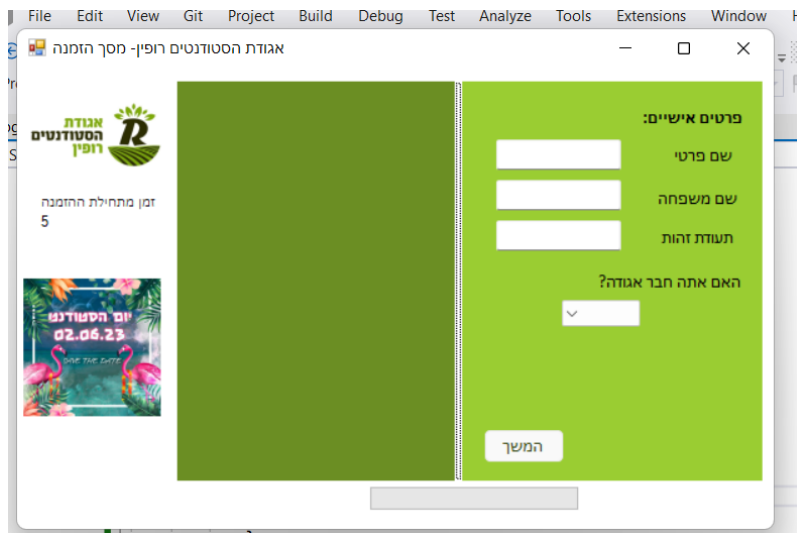
```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

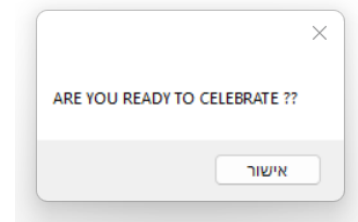
namespace StudentDay
{
    internal static class TicketsList
    {
        public static int numOfTickets;
        public static Ticket[] AddTicket(Ticket ticket, Ticket[]
ticketsList)
        {
            if (ticketsList.Length == 0)
            {
                ticketsList = new Ticket[1];
                ticketsList[0] = ticket;
                numOfTickets = 1;
                return ticketsList;
            }

            Array.Resize(ref ticketsList, numOfTickets + 1);
            ticketsList[numOfTickets] = ticket;
            numOfTickets++;
            return ticketsList;
        }
    }
}

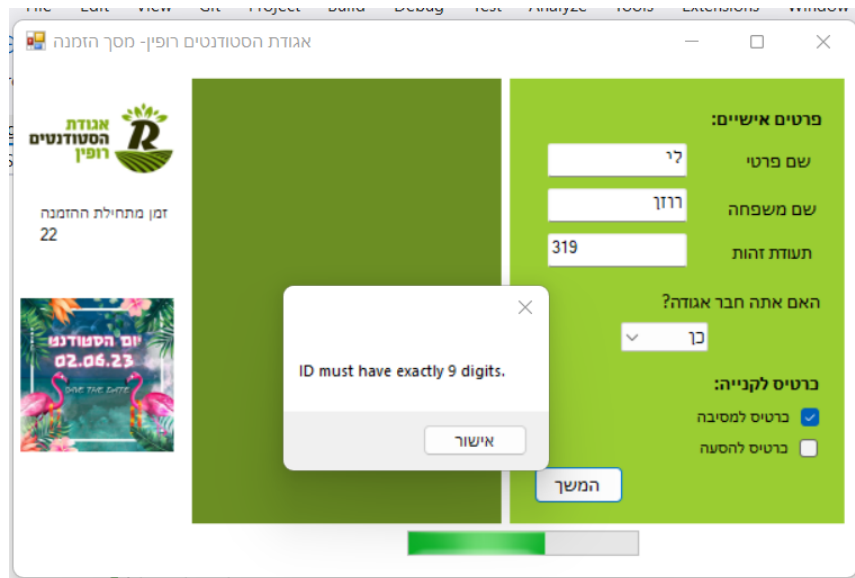
```



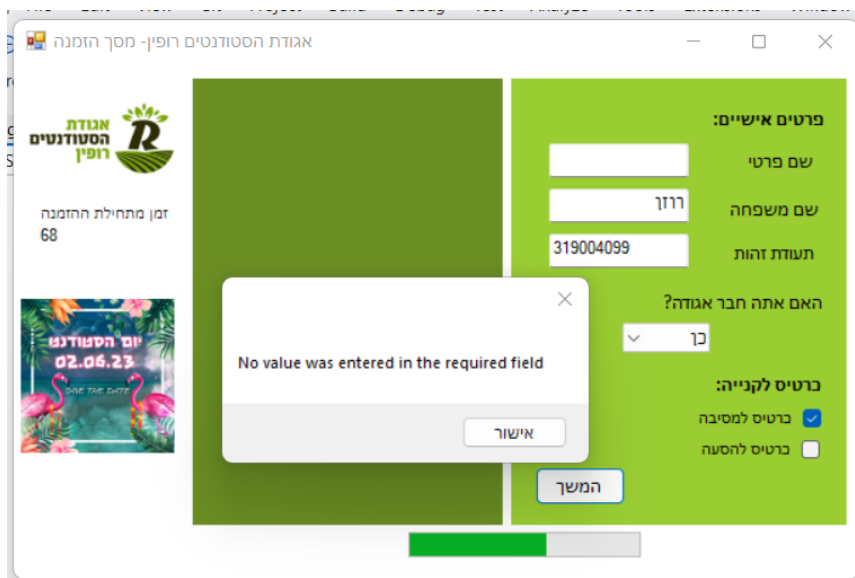
צילומי מסך:



הודעת שגיאה במקרה של מילוי שגוי של ת.ז.



הודעת שגיאה במקרה של אי מילוי פרטי שם פרטי או שם משפחה:



הודעת שגיאה במקרה שבחרים את התא הראשון הריק בחבר אגודה:
(אם לא בחרים בכלל אז הכפתור המשיך לא ניתן ללחיצה ואפשרות בחירת כרטיסים לא מופיעה)

The screenshot shows the 'Agoda R' registration form. On the left, there is a logo and text: 'אגודת הסטודנטים רופין' and 'זמן מתחילת החזמנה 33'. Below this is a promotional image for 'יום הסטודנט' on 02.06.23. The main form area is divided into two columns. The right column contains fields for 'שם פרטי' (First Name), 'שם משפחה' (Last Name), 'תעודת זהות' (ID Number), and a dropdown for 'האם אתה חבר אגודה?' (Are you a member?). The left column is mostly empty. A modal dialog box is displayed in the center with the title 'שגיאה במילוי הפרטים' (Error in filling details) and the message 'Please select your member option'. It has an 'אישור' (OK) button.

הממשק כאשר בחרים חבר אגודה "כן":

The screenshot shows the 'Agoda R' registration form after successful completion. The left column now contains two fields: 'מחיר כרטיס למסיבה:' (Party ticket price) with the value '64' and 'מחיר כרטיס להסעה:' (Transportation ticket price) with the value '8'. Below these is a button labeled 'אשר פרטים והזמן' (Approve details and book). The right column contains the same fields as before, but the 'האם אתה חבר אגודה?' dropdown is now set to 'כן' (Yes). Below this, there are two checked checkboxes: 'כרטיס לקנייה:' (Ticket for purchase) and 'כרטיס למסיבה' (Party ticket). A 'המשך' (Continue) button is visible at the bottom right.

אחרי לחיצה על כפתור "אשר פרטים והזמן":

אגודת הסטודנטים רופין- מסך הזמנה

אגודת הסטודנטים רופין

זמן מתחילת ההזמנה: 98

יום הסטודנטים 22.06.23

פרטים אישיים:

שם פרטי: לי
שם משפחה: רוזן
תעודת זהות: 319004099

מחיר כרטיס למסיבה: 64
מחיר כרטיס להסעה: 8

ביצוע תשלום

לידיעתך, התשלום על הרכישה יבוצע מפרטי כרטיס האשראי ששמורים בחשבונך באתר האגודה. לסיום הזמנה לחץ על כפתור סיום ההזמנה

אישור

אגודת הסטודנטים רופין- מסך הזמנה

אגודת הסטודנטים רופין

זמן מתחילת ההזמנה: 141

יום הסטודנטים 22.06.23

פרטים אישיים:

שם פרטי: לי
שם משפחה: רוזן
תעודת זהות: 319004099

האם אתה חבר אגודה?
כן

כרטיס לקנייה:

כרטיס למסיבה ☒
כרטיס להסעה ☒

אשר פרטים והזמן

המשך

סיום הזמנה

אגודת הסטודנטים רופין- מסך הזמנה

אגודת הסטודנטים רופין

זמן מתחילת ההזמנה: 161

יום הסטודנטים 22.06.23

פרטים אישיים:

שם פרטי: לי
שם משפחה: רוזן
תעודת זהות: 319004099

מחיר כרטיס למסיבה: 64
מחיר כרטיס להסעה: 8

סיכום ההזמנה שלך:

שם מלא: רוזן לי
תעודת זהות: 319004099
סוג כרטיס: מסיבה
מחיר: 64
סוג כרטיס: אוטובוס
מחיר: 8

זמן ביצוע ההזמנה: 18:16:15

סה"כ כרטיסים: 2
סה"כ לתשלום: 72

מחכים לראותכם!

סיום הזמנה

הממשק כאשר בוחרים חבר אגודה "לא":

אגודת הסטודנטים רפ"ן - מסך הזמנה



אגודת הסטודנטים רפ"ן

זמן מתחילת החזמנה
51



פרטים אישיים:

שם פרטי:

שם משפחה:

תעודת זהות:


האם אתה חבר אגודה?

ברטיס לקנייה:

ברטיס למסיבה ☒


ברטיס להסעה ☐

אגודת הסטודנטים רפ"ן - מסך הזמנה



אגודת הסטודנטים רפ"ן

זמן מתחילת החזמנה
79



מחיר ברטיס למסיבה:

פרטים אישיים:

שם פרטי:

שם משפחה:

תעודת זהות:

האם אתה חבר אגודה?


ברטיס לקנייה:

ברטיס למסיבה ☒

ברטיס להסעה ☐


לאחר לחיצה על כפתור "בחירת אופן תשלום":

אגודת הסטודנטים רפ"ן - מסך הזמנה



אגודת הסטודנטים רפ"ן

זמן מתחילת החזמנה
105



מחיר ברטיס למסיבה:

פרטים אישיים:

שם פרטי:

שם משפחה:

תעודת זהות:

האם אתה חבר אגודה?

ברטיס לקנייה:

ברטיס למסיבה ☒

ברטיס להסעה ☐

בחירת "תשלום בכניסה לאירוע":

* בלחיצה על כפתור "תשלום דרך ביט" נפתח קישור לאתר ביט

The screenshot shows a web application window titled "אגודת הסטודנטים רופין - מסך הזמנה". The main content area is divided into two panels. The left panel has a green background and contains a logo, a date "זמן מתחילת ההזמנה 164", a promotional image for "יום הסטודנטים 22.06.23", and two buttons: "תשלום בכניסה לאירוע" and "תשלום דרך ביט". The right panel has a light green background and contains a form titled "פרטים אישיים:" with fields for "שם פרטי", "שם משפחה", "תעודת זהות", "האם אתה חבר אגודה?", "ברטיס לקנייה:", and "ברטיס למסיבה:". A dialog box is open in the foreground with the title "בחירת אופן תשלום: תשלום במקום". It contains a warning icon and text: "שים לב, כניסתך לאירוע תהיה מותנית בביצוע התשלום בכניסה, ניתן במזומן או באשראי לבחירתך." At the bottom of the dialog are buttons for "אישור" and "ביטול".

This screenshot shows the same web application window, but the dialog box now displays a success message: "בחירת אופן תשלום: תשלום במקום". The message text reads: "בבחינתך התקבלה בהצלחה לסיום הזמנה לחץ על כפתור סיום ההזמנה". At the bottom of the dialog is a button labeled "אישור". The background form and buttons remain the same as in the previous screenshot.

