Ministerul Educatiei al Republicii Moldova
Universitatea Tehnica a Moldovei
Facultatea Calculatoare Informatica si Microelectronica
Departamentul Inginerii Software și automatica

Proiect de curs

Disciplina: APPOO

Tema: Elaborarea unei aplicatii Email Client

A efectuat: stud. gr. TI-142 Cojucari Dan

A verificat: lect.asis. Gavrișco Alexandru

Lect. Sup. Poștaru Andrei

Chișinău 2017

Cuprins

Introducere

Programarea orientată pe obiecte este o tehnologie modernă in domeniul programării calculatoarelor, care a rezultat din necesitatea realizării de aplicaţii din ce în ce mai complexe. Programarea clasica si structurata avea urmatoarele dezavantaje: control greoi al programelor de dimensiuni mari, dificultati cu privire la reutilizarea codurilor de programe si inflexibiliatea adaptarii si extinderii unor module de program deja realizate. Programarea Orientata pe Obiecte, POO, se fundamenteaza pe conceptul de obiect, care este definit printr-o multime de date, numite proprietati, si o multime de proceduri sau functii de prelucrare ale acestor date, numite metode.

Programarea orientată pe obiecte (object-oriented programming) este o metodă de programare în care programele sunt organizate ca şi colecţii de obiecte cooperante, fiecare dintre ele reprezentând o instanţă a unei clase, iar clasele sunt membre ale unei ierarhii de clase, corelate între ele prin relaţii de moştenire.

Există trei părţi importante ale acestei definiţii:

- Se folosesc obiecte, nu algoritmi, ca unităţi constructive de bază , obiectele fiind componente ale unei ierarhii de agregare.
- Fiecare obiect este o instanţă (un exemplar) al unei clase.
- Clasele sunt componente ale unei ierarhii de tip, fiind corelate între ele prin relaţii de moştenire.

Ideea POO este de a crea programele ca o colecţie de obiecte, unităţi individuale de cod care interacţionează unele cu altele, în loc de simple liste de instrucţiuni sau de apeluri de proceduri.

Obiectele POO sînt de obicei reprezentări ale obiectelor din viaţa reală, astfel încît programele realizate prin tehnica POO sînt mai uşor de înţeles, de depanat şi de extins decît programele procedurale (mai ales în cazul proiectelor software complexe şi de dimensiuni mari, care se gestionează făcînd apel la ingineria programării).

1.Sarcina tehnica a sistemului

Pentru implementarea sistem-ului voi folosi Limbajul de programare C# și platforma *.Net*

Toate conexiunile dintre aplicație și server vor fi securizate, folosind protocolul *SSL*, pentru aceasta se va folosi clasa *SslStream*, care primind stream-ul conexiunii TCP, îl înlocuiește cu unul securizat.

Interfata aplicației va fi creată folosind *WindowsForms*.La pornirea aplicației utlizatorului i se va afișa o fereastră un va trebui să introducă configurările necesare pentru a avea acces la email acestea sunt: adresa IMAP a serverului și portul acesteia, adresa SMTP a serverului și portul respectiv.Dupa introducerea acestor date utlizatorul va trebui să introduca adresa de *Email* și *Parola*.Dupa introducerea tuturor datelor utilizatorul va trebui sa apese butonul Autentificare pentru a intra pe email-ul sau.

După autentificare utlizatorului si se va afișsa o fereastră unde va fi afișată lista de email-uri.Apasînd pe subiectul unui email utlizatorului i se va deschide o fereastră unde va arăta conținutul acestuia.Pentru trimiterea unui email nou tot in această fereastră va fi un buton pe utlizatorul va trebui sa acceseze.Utilizatorului i se va deschide altă fereastră unde va trebui să introducă email-ul destinatar-ului,subiect-ul email-ului și conținut-ul acestuia.Daca utlizatorul va avea de trimis fișiere acesta va trbui sa apese butonul *Adaugare fișier* după care va trebui să însereze din calculator fisșierul dorit.

Pentru a ieși de pe email va fi și un buton de ieșire pentru ca utlizator-ul sa se delogheze și respectiv se va putea autentifica pe un alt email.

2.Diagrame uml

UML (Unified Modelling Language) reprezinta un limbaj vizual de modelare folositor în domeniul software, dedicat construirii sistemelor complexe si realizarii documentelor de specificaţii, facand referire in mare parte la vizualizarea, specificarea, construirea şi documentarea sistemelor de aplicaţii. Prezinta si limitări cu privire la generarea codului şi reprezinta de asemenea un mijloc bun pentru domeniul ingineriei programării.

Scopul unui limbaj de modelare este analiza si proiectarea programelor. UML reprezinta limbajul universal standard pentru dezvoltatorii software de pretutindeni, si de asemenea o combinatie excelenta a celor mai bune trei limbaje de modelare anterioare orientate pe obiecte (Booch, OMT, and OOSE). Asadar limbajul UML reuneşte cele mai bune tehnici şi practici din domeniul ingineriei programării, care şi-au dovedit eficienţa în construirea sistemelor complexe, rezultatul avand o expresivitate foarte buna care ajuta la rezolvarea diverselor probleme de modelare pe care vechile limbaje nu reuseau sa le indeplineasca foarte bine. UML ar putea indeplini pe langa rolul de limbaj vizual de modelare si cel de limbaj vizual de programare, dar momentan nu dispune de întreg sprijinul semantic şi vizual pentru a înlocui limbajele de programare.

2.1.Diagramele Use Case

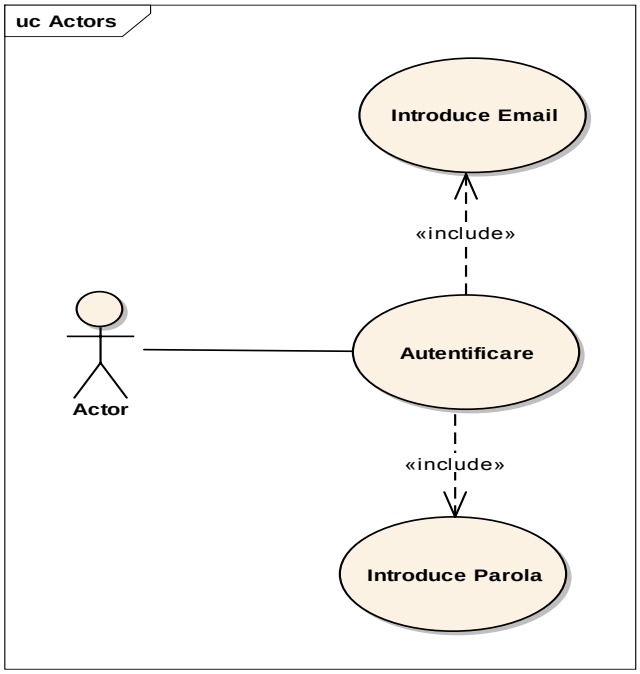

Figura 1. Diagrama caz de utilizare pentru autentificare

Dupa Introducerea configurărilor utlizatorul trebuie să introducă email-ul și parola al email-ului lacare vrea să se conecteze.reprezentat în *fig. 1*.

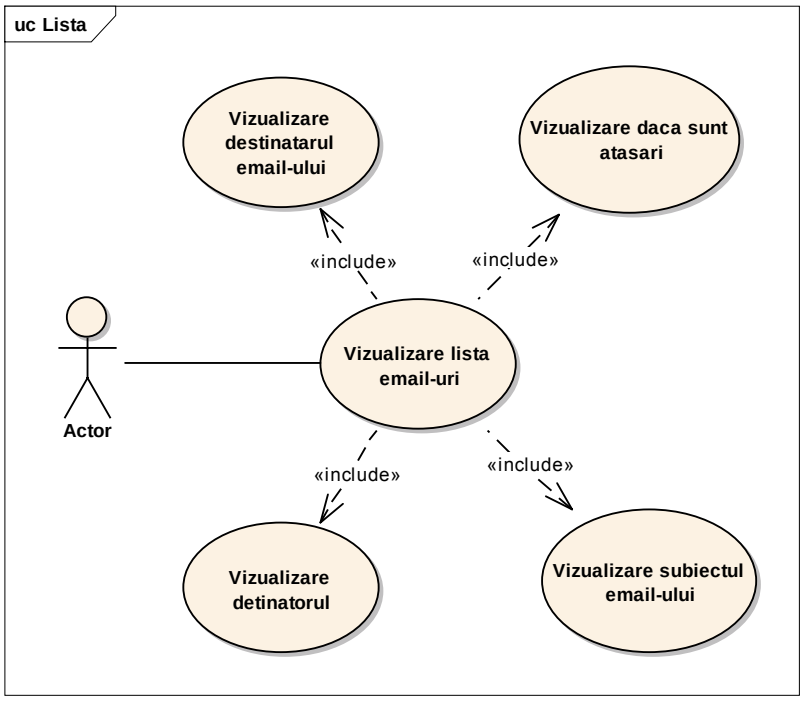

Figura 2. Diagrama caz de utilizare pentru vizualizare email-urilor

În *figura 2* am reprezentat diagramama Use Case pentru modul in care sunt vizualizate email-urile.Email-urile sunt împarțite în mape.Cum ar fi mesaje (*Primite, Spam,*

Trimise,etc.). Lista de email-uri ne oferă posibilitatea de a vedea cine a trimis email-ul, cine a primit, subiect-ul email-ului.Se poate vizualiza chiar daca a fost trimis unui grup de emai-uri.

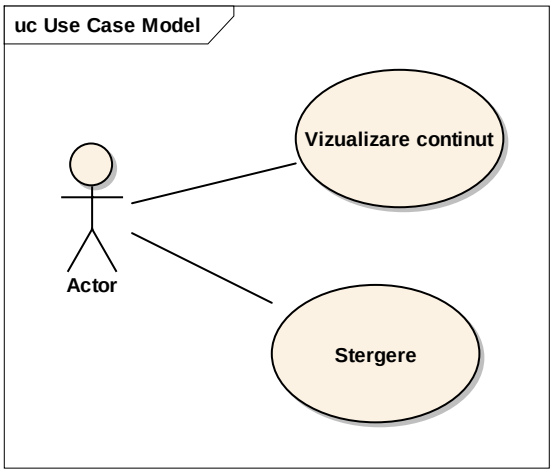

Figura 3. Diagrama caz de utilizare pentru acțiuni asupra email-ului

Dupa ce am vizualizat lista de email-uri putem accesa un email și să vedem conținutul acestuia.În dreptul fiecărui email este butonul de ștergere.Utilizatorul poate zualiza conținutul eail-ului și dupa asta daca dorește poate să îl șteargă. *Fig.3*

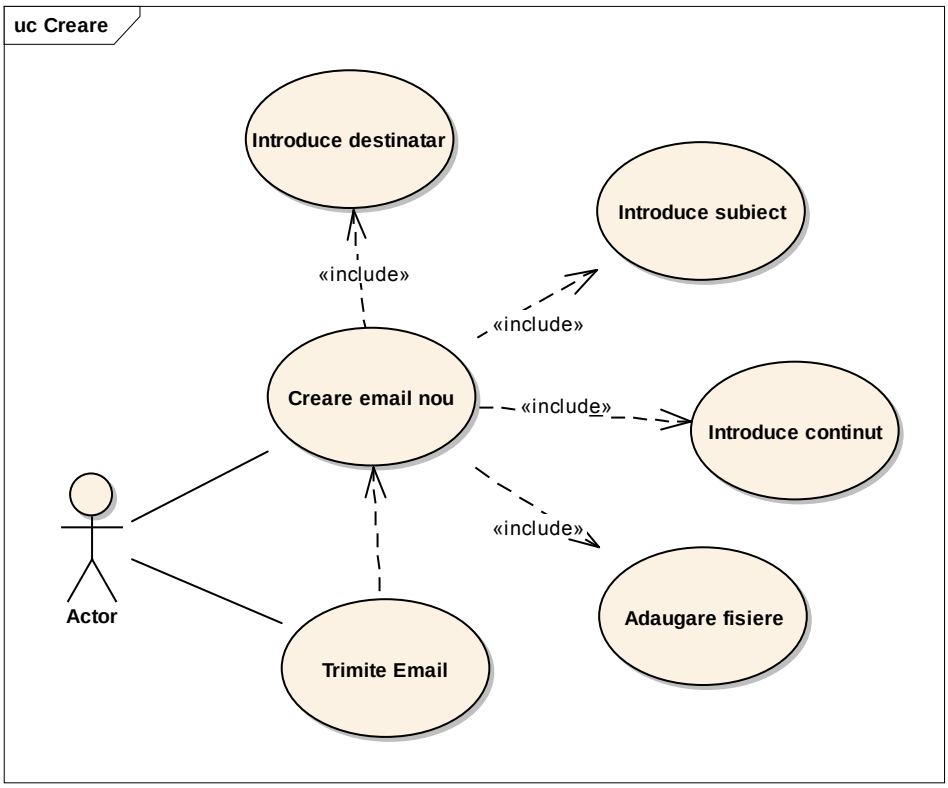

Figura 4. Diagrama caz de utilizare pentru trimiterea unui email

Pentru a trimite un email utlizatorul trebuie sa acceseze butonul *Trimite Email*.Pentru a trimite un email utlizatorul va trebui să introducă adresa de email a destinatarului, subiectul email-ului, conținut-ul email-ului și daca vrea să trimită fișiere trebuie să apese butonul *Adaugă Fișier* dupa aceasta poate alege un fișier din calculator și să-l

trimită.Trimiterea email-ului nu se va efectua daca nu va fi introdus email-ul destinatarului. *Fig.4.*

2.2.Elaborarea diagramelor de claase

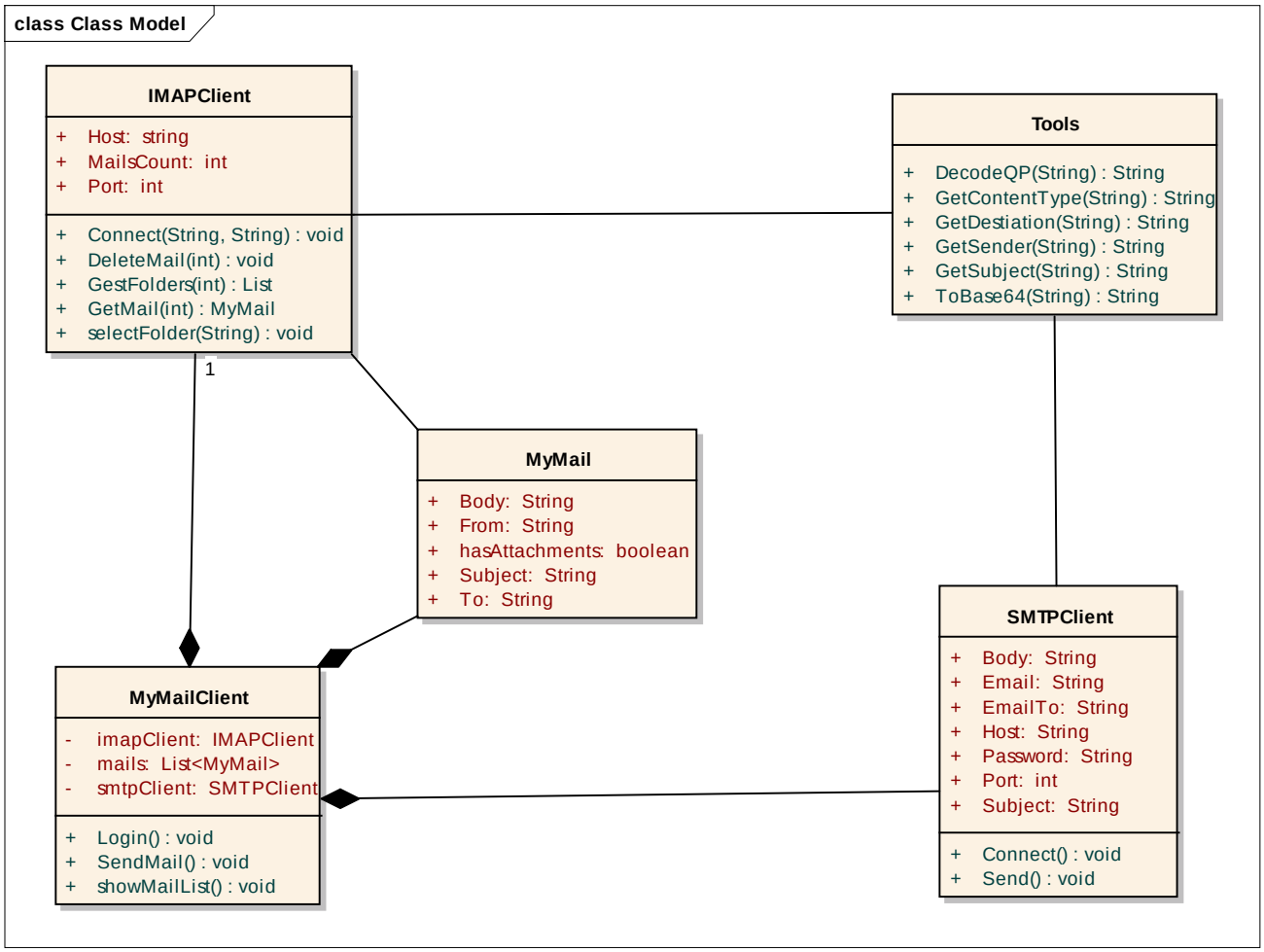

Figura 5. Diagrama de clase

Proiectul conține mai multe clase dar eu am cread doar o singură doar o singură diagramă.Clasa de baza a proiectului este *MyMailClient*, aceasta conține cîte un obiect al claselor *SMTPClient* și *IMAPCLient*, și o colecție de obiecte de tip *MyMail*. De asemeni clasa *MyMailClient* conține și elemente de interfață a aplicației, care nu au fost introducse în diagrama de clase, deoarece nu joaca rol în funcționarea logică a aplicației. Clasa *MyMail* nu are metode, ci doar atribute. Această clasă se folosește pentru stocarea datelor email-ului(subjet, body, from, to).

Clasa *Tools* conține doar metode, ea este formată dintr-o colecție funcții(codificatoare, decodificatoare, parsere) folosite de alte clase.

Clasa *SMTPClient* este folosită pentru transmiterea email-urilor. Aceasta are doar 2 metode funcționale: conectarea la server, si transmiterea email-ului.

Clasa *IMAPClient* oferă posibilitatea de extragere a email-urilor dintr-o mapă anumită.

Funcționalitățile acesteia sunt:

- conecatarea la server;
- extragerea listei de mape în căsuța poștală;
- selectarea unei mape;

- extragerea email-urilor din mapa selectată;
- ștergerea unui email

2.3.Elaborarea diagramelor de secvență

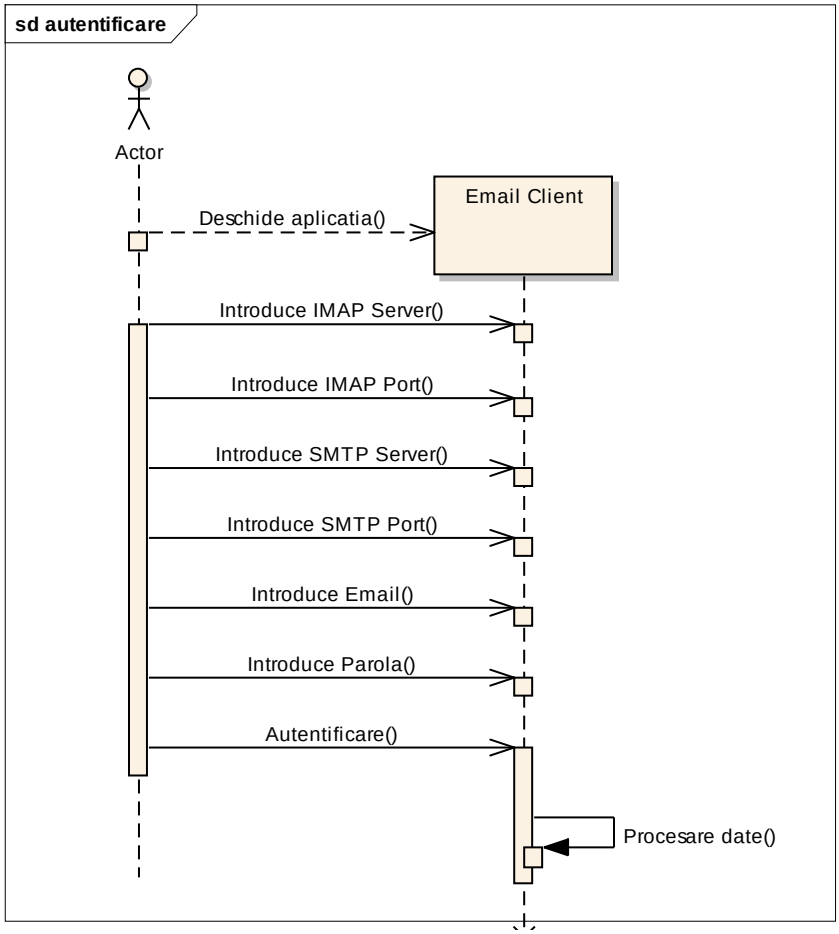

Figura 6. Diagrama de secvență pentru autentificare

În *figura 6* am reprezentat modul în care se poate autentifica un utlizator în sistem.La pornirea aplicației utlizatorul trebuie să introducă configurările necesare pentru pentru emai-ul la care vrea să se concteze.
Acestea sunt:

- adresa server-ului IMAP și portul acestuia
- adresa server-ului SMTP și respectiv portul acestuia

Dupa introducerea configurărilor utlizatorultrebuie să introduca adresa de email si parola pentru a se autentifica.

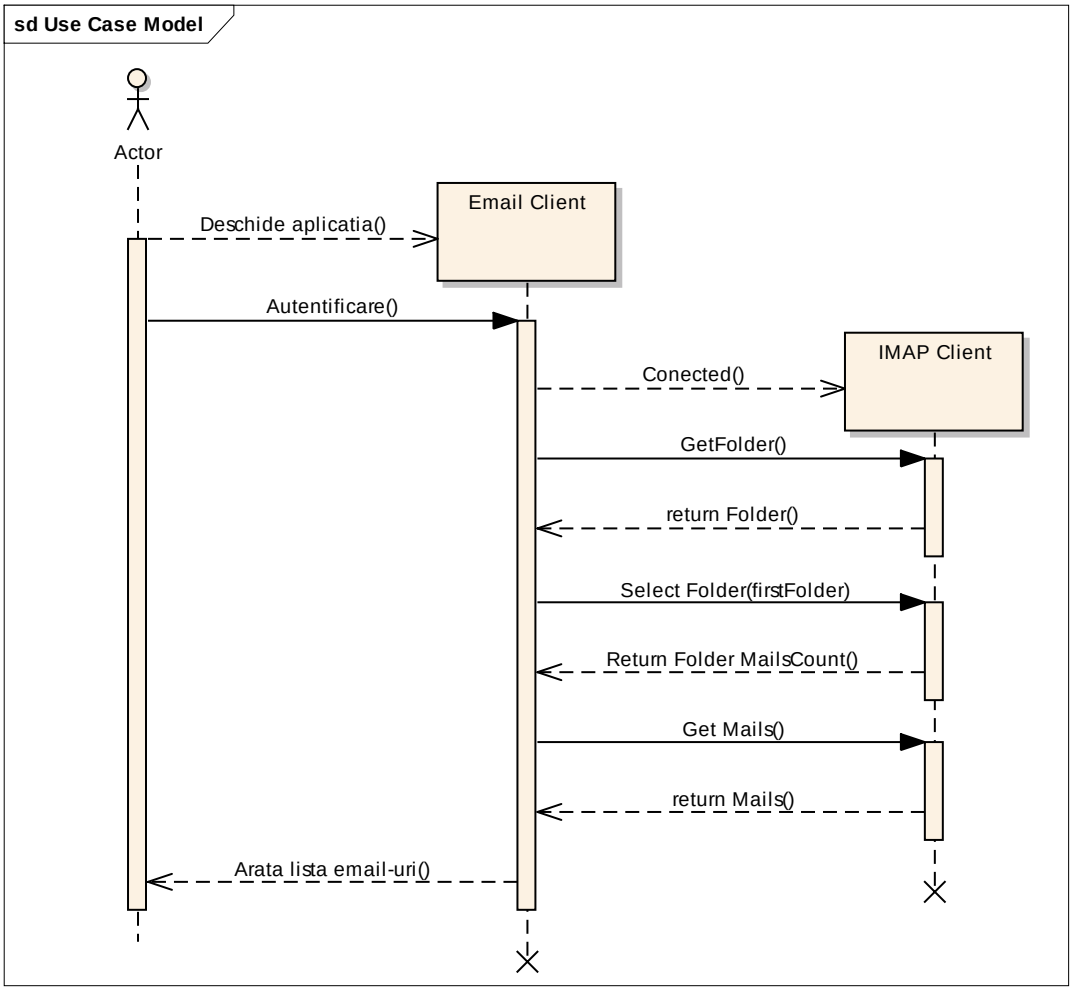

Figura 7. Diagrama de secvență pentru afișarea email-urilor

Fig.7 arată cum are loc afișarea listei email-urilor. După autentificare, aplicația prin intermediul clasei se conectează la server, cere lista de mape și o fișează utilizatorului. Predefinit la pornirea aplicației, se selectează prima mapă din listă și se extrag email-urile din acea mapă. După asta utilizatoarul poate selecta altă mapă, și se vor extrage email-urile deja din mapa nou aleasă.

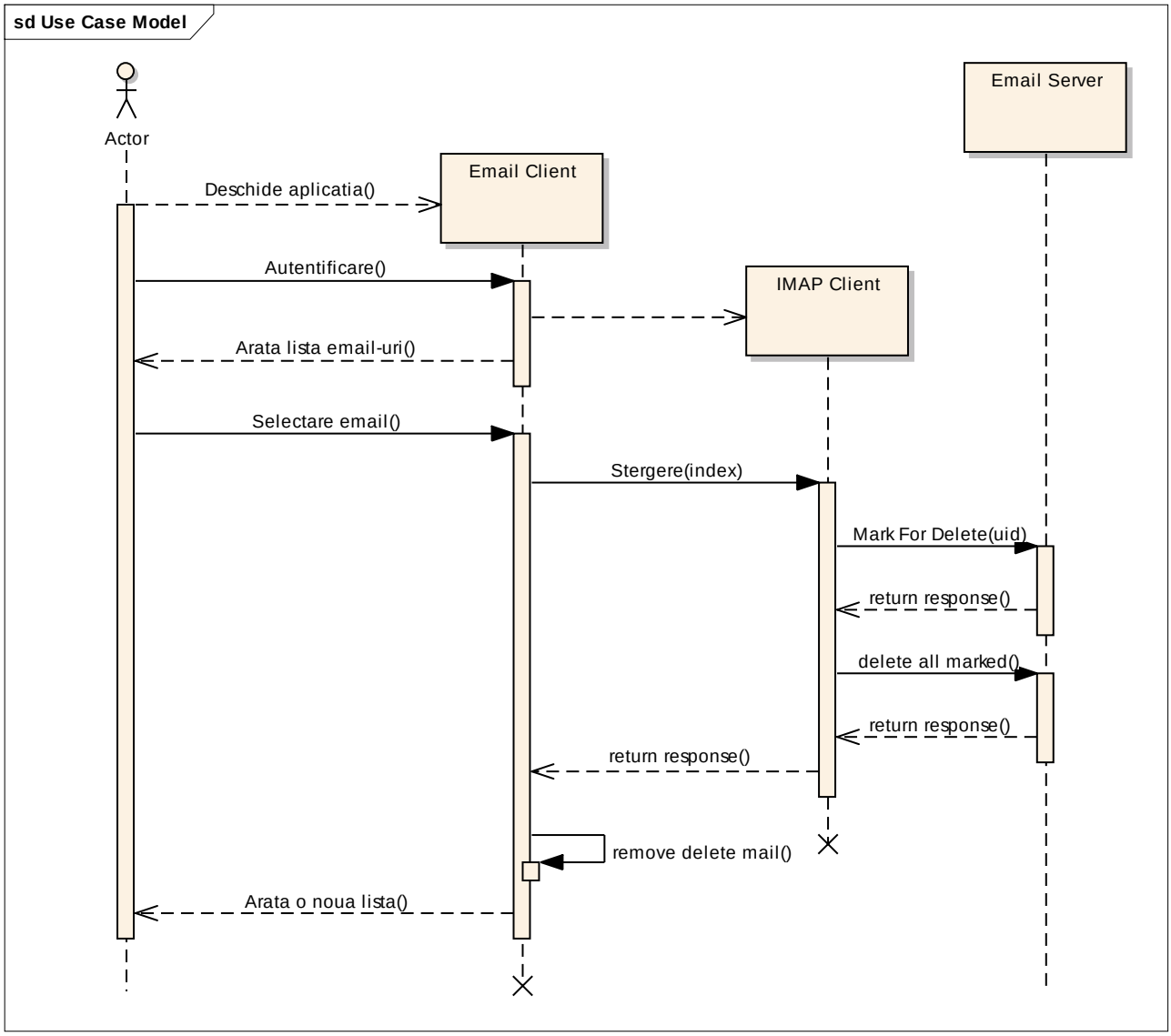

Figura 8. Diagrama de secvență pentru stergerea unui email

Utilizatorul are posibilitatea de a șterge un email, modul de ștergere este reprezentat de diagrama. După autentificare și afișarea listei email-urilor, utilizatorul poate șterge email-ul prin apăsarea butonului „Ștergere”. După apăsarea butonului, prin intermediul clasei, se transmite la server un mesaj pentru marcarea email-ului pentru ștergere, după care se transmite alt mesaj, pentru ștergerea tuturor mesajelor marcate pentru ștergere.

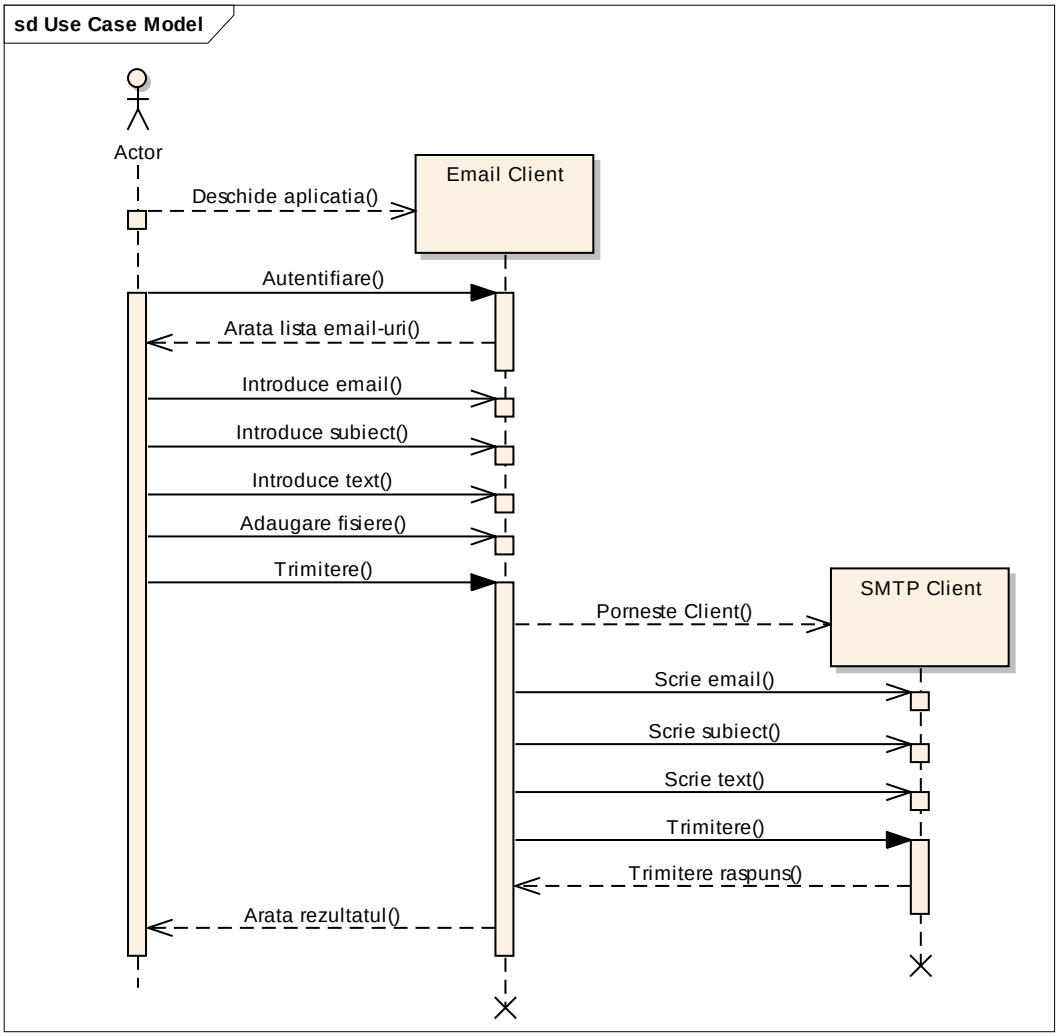

Figura 9. Diagrama de secvență pentru trimiterea unui email

Pentru a trimite un email, utilizatorul trebuie să introducă adresa destinatarului, subiectul email-ului, conținut-ul email-ului și respectiv poate adăuga și fișiere. După introducerea tuturor datelor, trebuie să apese butonul „*Trimitere*”, dacă nu a fost introdus email-ul destinatar-ului, butonul nu v-a funcționa. La apăsarea butonului, prin intermediul SMTPClient, se face o conexiune la server, se încarcă toate datele în SMTPClient, și se transmit la server conform protocolului. În caz că email-ul a fost transmis cu succes, utilizatorului i se va afișa un mesaj despre acesta. Aceasta este reprezentat în diagrama de secvență din *fig.9*.

2.4.Elaborarea diagramelor de activități

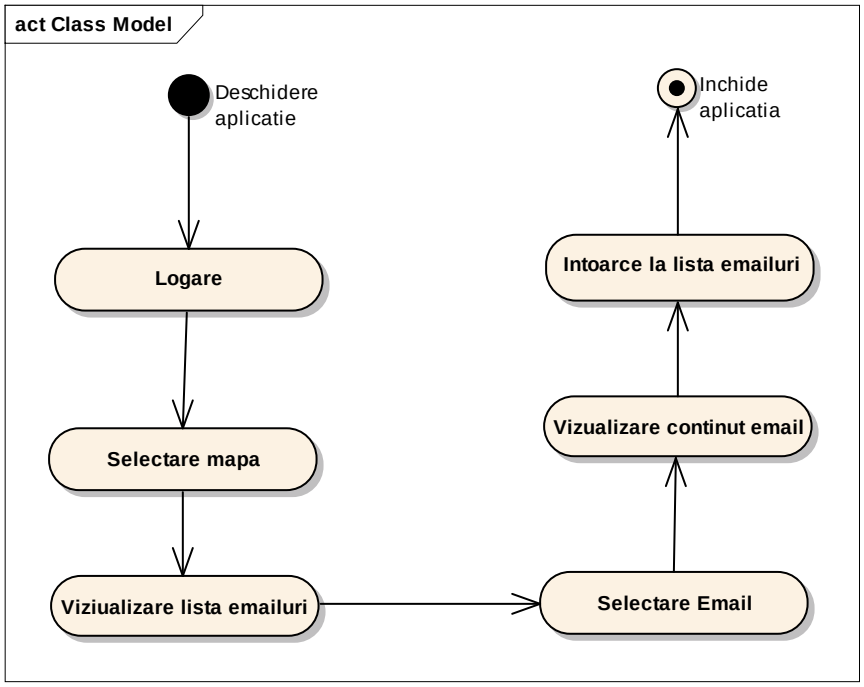

Figura 10. Diagrama de activitate pentru vizualizarea unui email

Pentru a putea vizualiza conținutul unui email utilizatorul trebuie să execute un set de acțiuni, pentru a arăta aceste acțiuni, am elaborat diagrama de activități, care este prezentată de *fig.10*. Începînd cu deschiderea aplicației, utilizatorul trebuie să se auntentifice, să selecteze mapa în care se află email-ul dorit, să-l selecteze din listă, după care poate să vizualize conținutul acestuia.

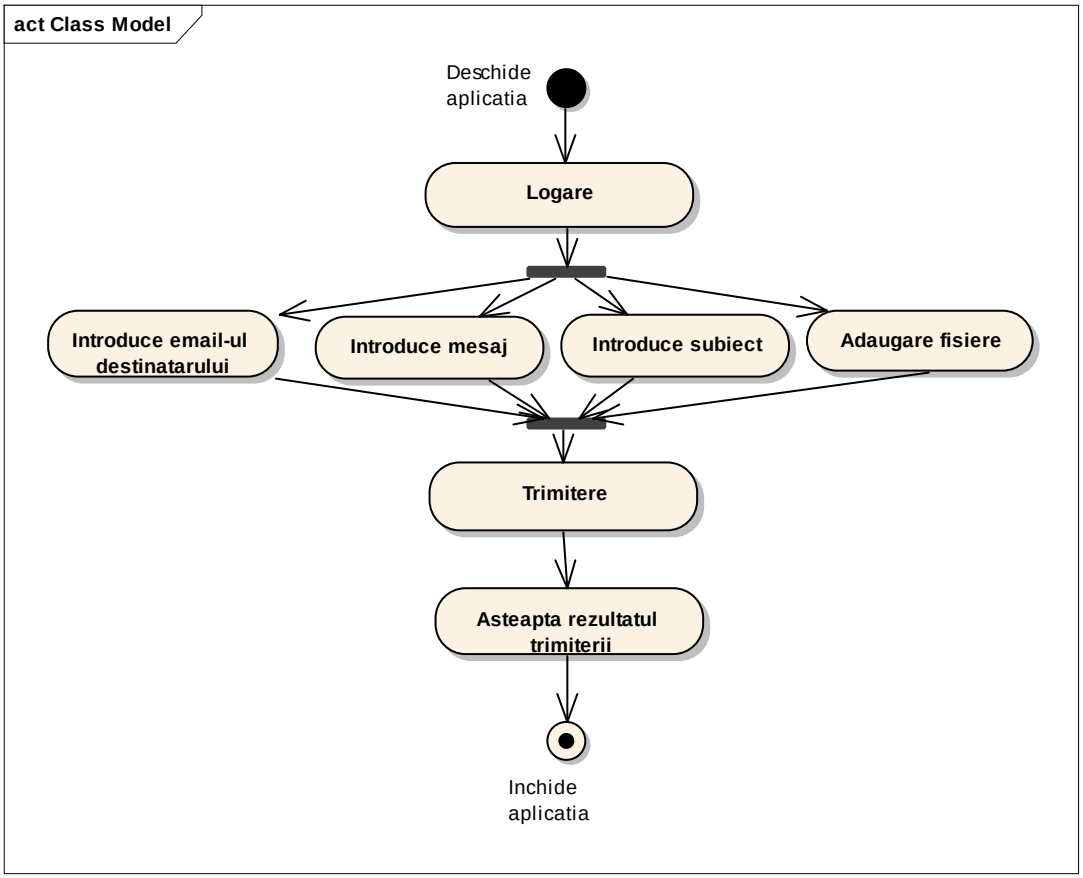

Figura 11. Diagrama de activitate pentru trimiterea unui email

Transmiterea unui email, de asemeni, necesită un set de activități executate de utilizator. După deschiderea aplicației și autentificarea, acesta trebuie să introducă datele email-ului, nu neapărat într-o ordine aumită, dar obligatoriu pe toate. Activitățile date sunt reprezentate de diagrama de activități din *fig.11*.

2.5.Elaborarea diagramelor de stări

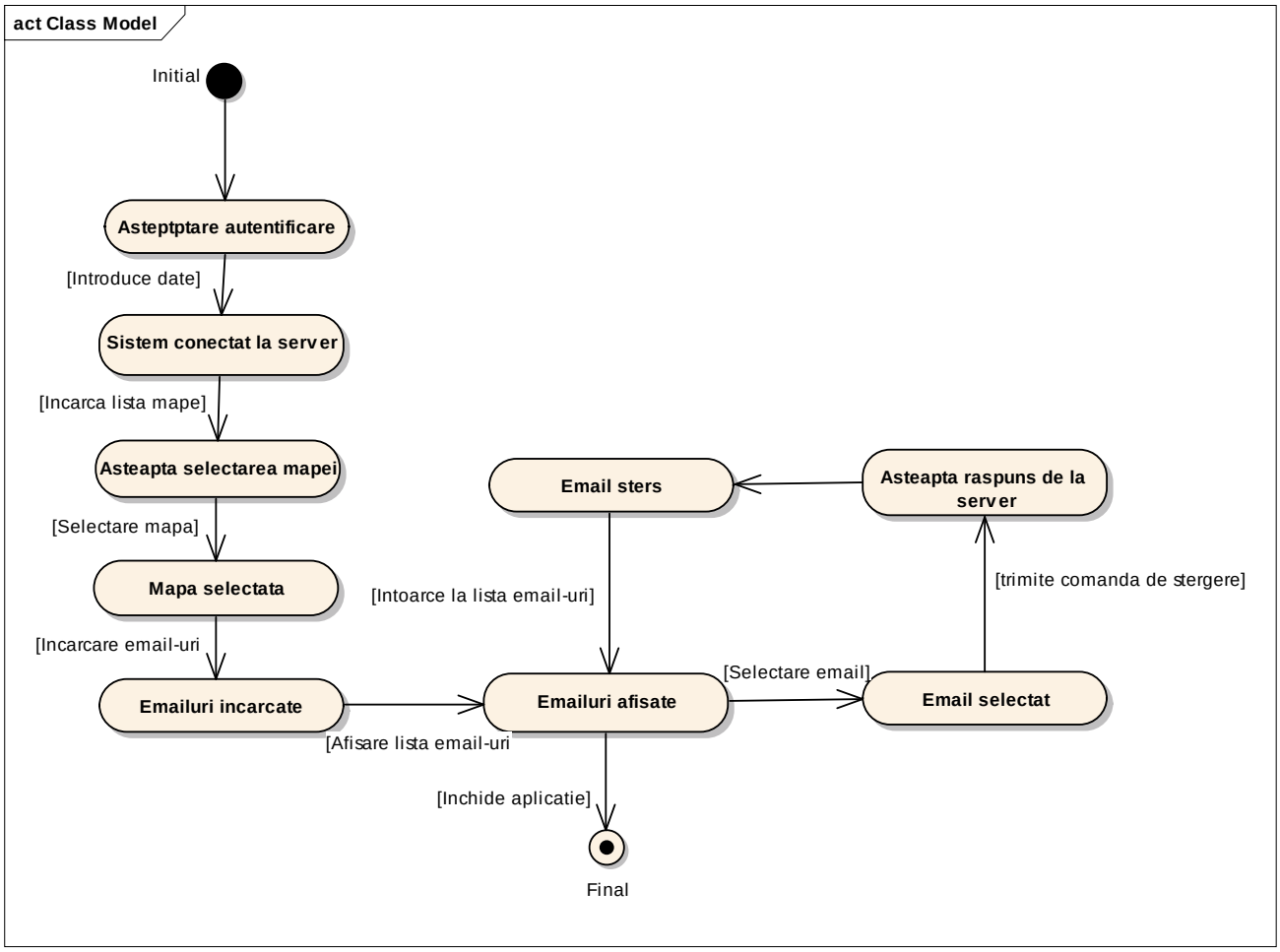

Figura 12. Diagrama de stari: stergerea unui email

În cadrul execuției, sistemul trece prin mai multe stări. *Fig.12* arată stările prin care trece sitemul în timpul stergerii unui email. După pornirea aplicației, se așteaptă ca utilizatorul să introducă datele de autentificare, după aceasta se conectează la server și se află în starea respectivă. După ce încarcă lista mapelor, sistemul așteaptă ca utilizatorul să selecteze mapa dorită. Cînd email-urile sunt afișate, sistemul se află în așteptarea selectarii email-ului de către utilizator. La selectarea email-ului se trimite comanda la server, pentru a-l șterge. După ce email-ul a fost șters, sistemul se întoarce la starea cînd este afișată lista cu email-uri, și așteaptă ori selectarea altui email, ori închiderea aplicației.

2.6.Elaborarea diagramelor de componente

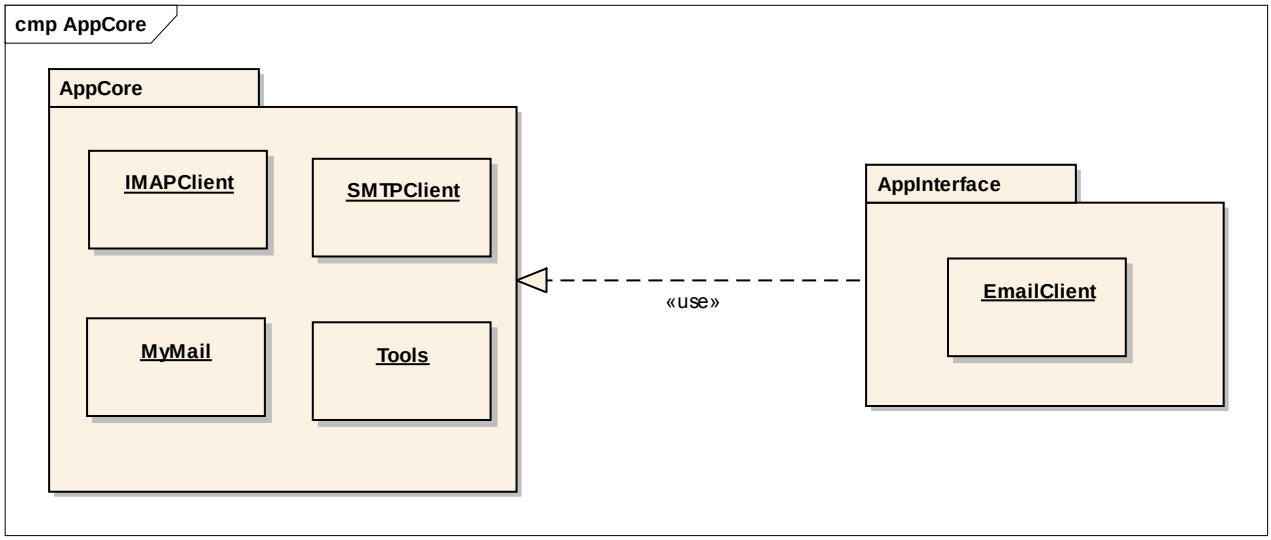

Figura 13. Diagrama de componente a sistemului

Diagrama de componente aratăcomponentele sistemului și modul în care sunt organizate acestea.
Acestea sunt:

- *AppInterface*: interfața aplicației și toate componentele de interacțiune cu utilizatorul;
- *AppCore*: clasele ce cuprind logica sistemului
  - SMTPClient;
  - MyMail;
  - Tools.

Interfața aplicației

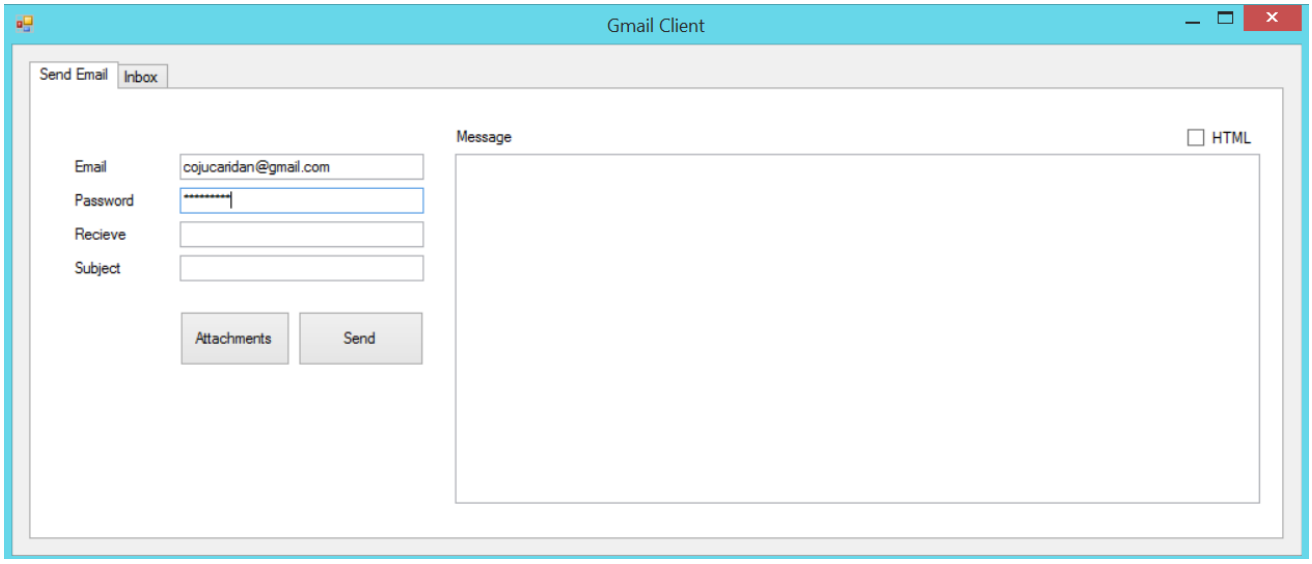

Fig 14.Pagina de start a aplicației

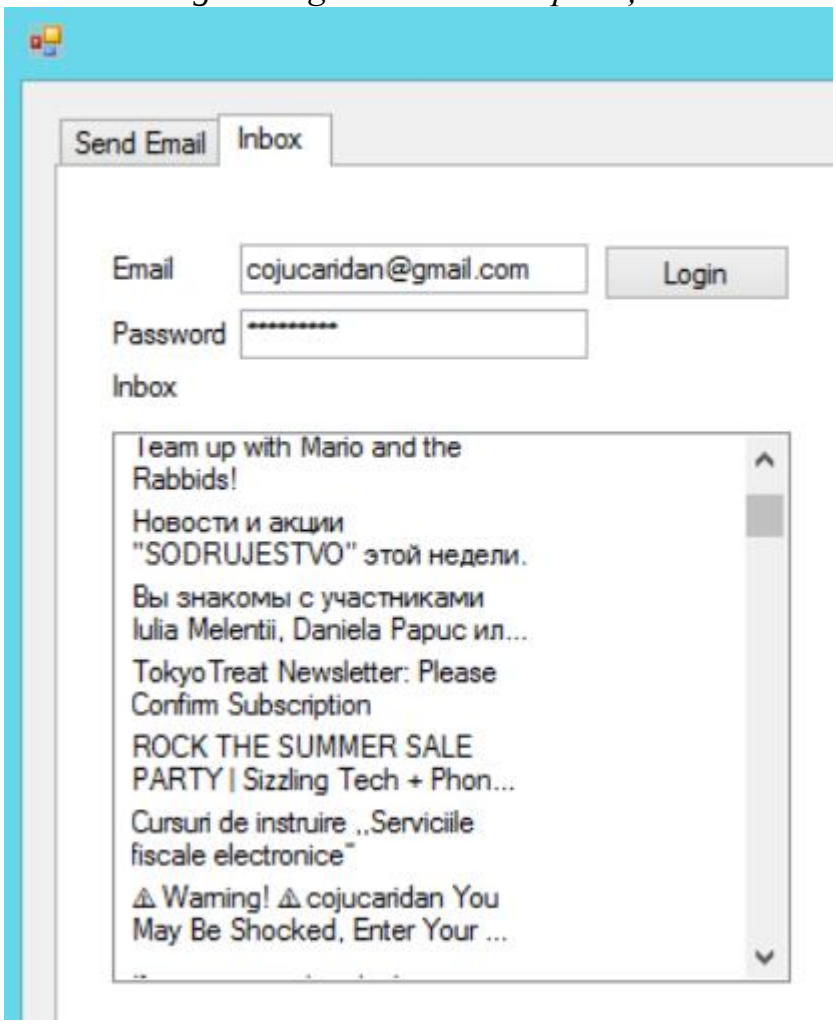

Fig.15. Fereastra cu lista de email-uri

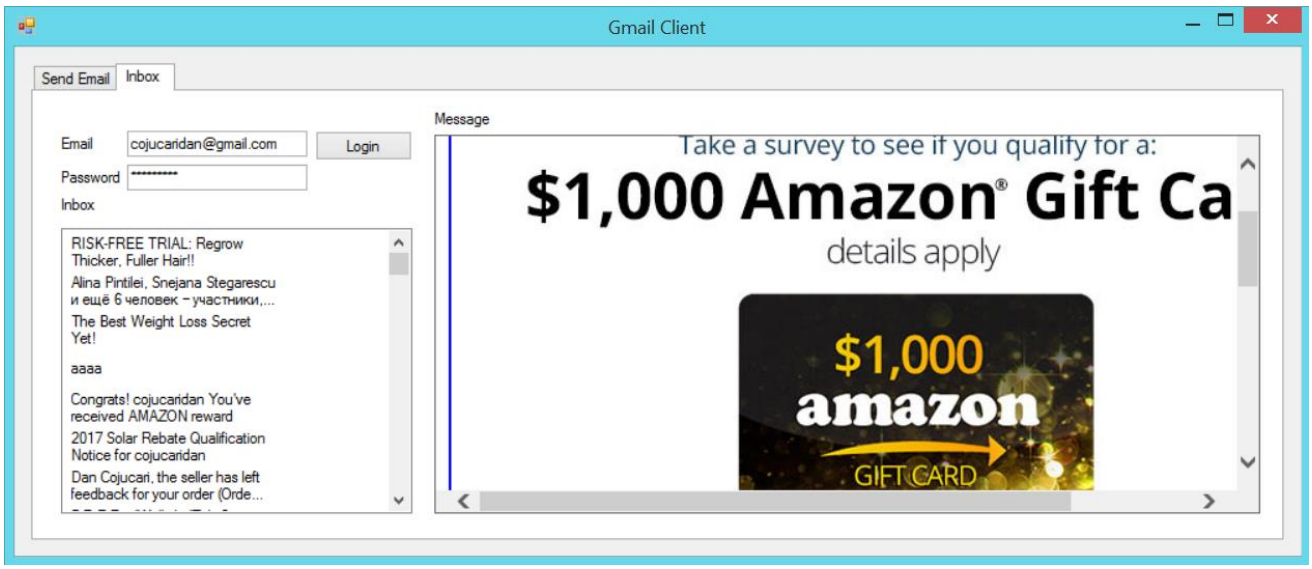

Fig.16. Fereastra în care este arătat conținut-ul unui email

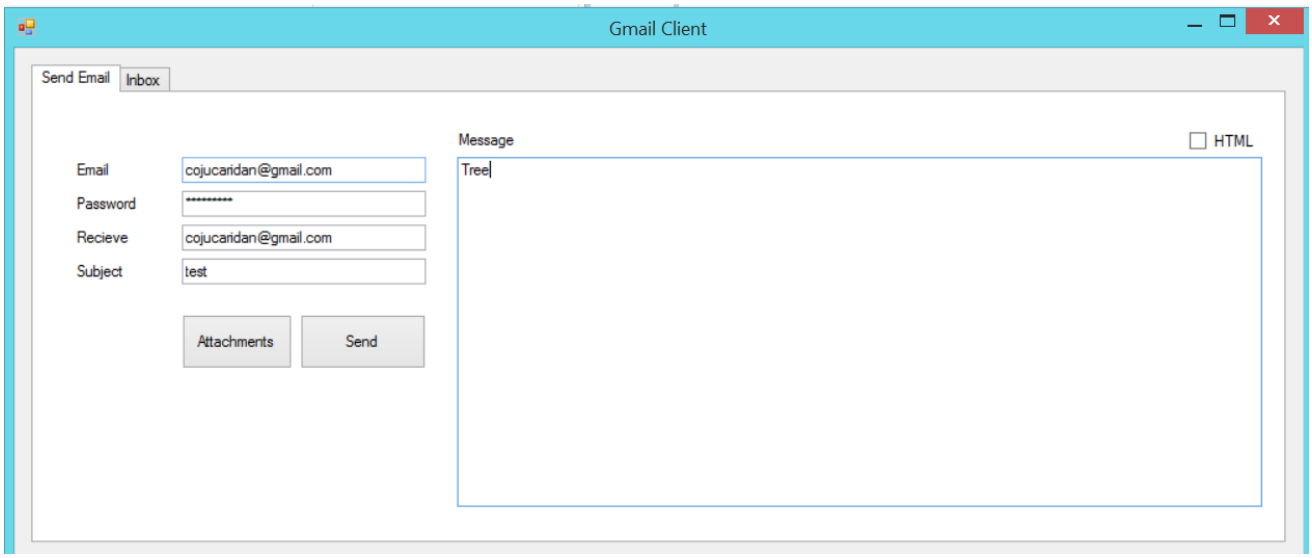

Fig.17. Fereastra în care trimitem un email

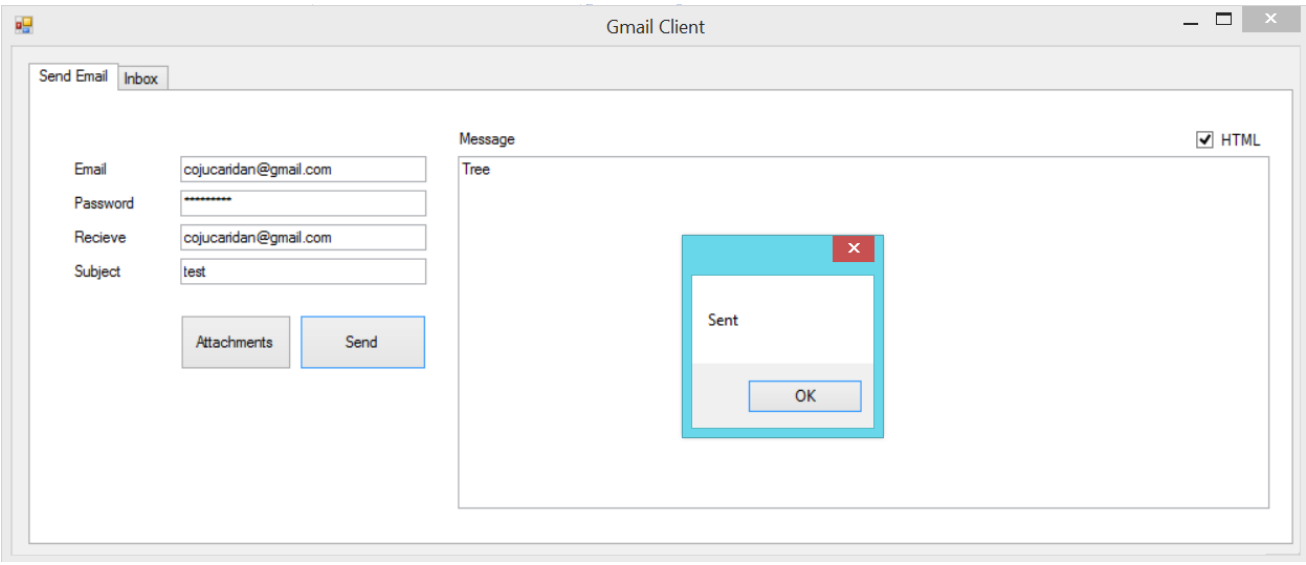

Fig.18. Fereastra în care confirm trimiterea emailului

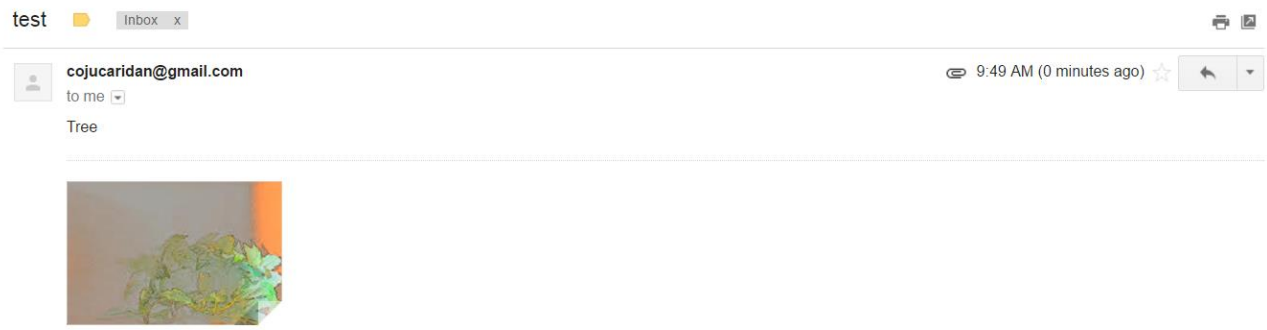

Fig.19. Fereastra în care confirm primirea emailului

Concluzie

Efectuind aceast proiect am realizat diagramele caz de utilizare pentru a modela sistemul propus, *Mail Client*. Cu ajutorul diagramelor caz de utilizare am specificat acțiunile pe care le poate executa utilizatorul asupra unui email, sau acțiunile care trebuie numaidecît executate. Pentru a reprezenta destinația funcțională a sistemului am elaborat diagrama cazurilor de utilizare, iar pentru a reprezenta interacțiunea în timp și spațiu dintre obiectele ce participă la formarea sistemului, am elaborat diagrama de interacțiune compusă din diagrame de secvența. În cazul în care am descris structura internă a sistemului, am realizat diagrama claselor. Diagrama de stare am realizat-o cu scopul de a evidenția comportamentul obectelor pe parcursul ciclurilor sale de viața, specificînd stările în care obiectele pot trece și tranzițiile respective dintre aceste stări. Am reprezentat arhitectura sistemului prin deretminarea dependențelor dintre componente.

Anexa A – Form1.cs

```
using System;
using System.Linq;
using System.Net;
using System.Net.Mail;
using System.Windows.Forms;
using MailAddress = System.Net.Mail.MailAddress;
using System.Threading.Tasks;
using OpenPop.Pop3;
using System.Collections.Generic;
using System.IO;

namespace labPR4
{
    public partial class Form1 : Form
    {
        List<OpenPop.Mime.Message> ListMessages;
        List<string> Atasamente;
        public Form1()
        {
            InitializeComponent();
            Atasamente = new List<string>();
            ListMessages = new List<OpenPop.Mime.Message>();
        }
        private void button3_Click(object sender, EventArgs e)
        {
            try
            {
                var smtpServer = new SmtpClient("smtp.gmail.com");
                var mail = new MailMessage();

                mail.From = new MailAddress(myEmail.Text);
                mail.To.Add(recieveEmail.Text);
                mail.Subject = mySubject.Text;
                mail.Body = txtbody.Text;
                smtpServer.Port = 587;
                smtpServer.UseDefaultCredentials = false;
                smtpServer.EnableSsl = true;
                if (!(Atasamente.Count == 0))
                {
                    Atasamente.ForEach(element =>
                    {
                        var _attachment = new Attachment(element);
                        mail.Attachments.Add(_attachment);
                    });
                }
                    if (checkBox1.Enabled)
                    {
                        mail.IsBodyHtml = true;
                        mail.Body ="<html><head></head><body>"+ mail.Body + "</body></html>";
                    }
                    else
                    {
                        mail.IsBodyHtml = false;
                    }
                smtpServer.Credentials = new NetworkCredential(myEmail.Text,
myPassword.Text);
                smtpServer.SendAsync(mail, new object());
                smtpServer.SendCompleted += (obj, ew) => { MessageBox.Show("Sent"); };
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
```

```
        private async void button1_Click(object sender, EventArgs e)
        {
            var listMessages = await GmailsRetrievePOPAsync(inboxEmail.Text,
inboxPassword.Text);

            listView1.View = View.Tile;

            listMessages.ForEach(element =>
            {
                listView1.Items.Add(!element.Headers.Subject.Equals(string.Empty) ?
element.Headers.Subject : "(no subject)~" + element.Headers.Date);
            });

            ListMessages = listMessages;
            listView1.Click += ListView1_Click;
        }

        private void ListView1_Click(object sender, EventArgs e)
        {
            var mail = ListMessages.FirstOrDefault(x =>
x.Headers.Subject.Equals(listView1.SelectedItems[0].Text));
            if (mail == null)
            {
                mail = ListMessages.FirstOrDefault(x =>
x.Headers.Date.Equals(listView1.SelectedItems[0].Text.Split('~').ElementAt(1)));
            }
            OpenPop.Mime.MessagePart BodyString;
            if (mail.FindFirstHtmlVersion() != null)
                BodyString = mail.FindFirstHtmlVersion();
            else
                BodyString = mail.FindFirstPlainTextVersion();
            const string _fileSaveMail = "lastMail.html";
            BodyString.Save(new FileInfo(_fileSaveMail));

            string _savedInstance = string.Empty;

            using (var file = new FileStream(_fileSaveMail, FileMode.Open))
            {
                using (var stream = new StreamReader(file))
                {
                    _savedInstance = stream.ReadToEnd();
                }
            }

            webBrowser1.DocumentText = _savedInstance;
        }

        public static Task<List<OpenPop.Mime.Message>> GmailsRetrievePOPAsync(string Email,
string Password)
        {

            return Task.Run(() =>
            {
                List<OpenPop.Mime.Message> emailList = new List<OpenPop.Mime.Message>();
                //stream
                using (Pop3Client pop3Client = new Pop3Client())
                {
                    pop3Client.Connect("pop.gmail.com", 995, true);
                    try
                    {
                        pop3Client.Authenticate("recent:" + Email, Password,
AuthenticationMethod.UsernameAndPassword);
                    }
                    catch (Exception ex)
```

```
                    {
                        MessageBox.Show(ex.Message, "Error");
                        return emailList;
                    }
                    int MailsCount = pop3Client.GetMessageCount();
                    for (var i = MailsCount; i > 0; i--)
                    {
                        emailList.Add(pop3Client.GetMessage(i));
                    }
                }
                return emailList;

            });

        }

        private void buttonAttachments_Click(object sender, EventArgs e)
        {
            OpenFileDialog _filesDialog = new OpenFileDialog();
            if (_filesDialog.ShowDialog() == DialogResult.OK)
            {
                Atasamente.Add(_filesDialog.FileName);
            }
        }

        private void label3_Click(object sender, EventArgs e)
        {

        }
    }
}
```

Anexa B – Program

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

namespace labPR4
{
    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());
        }
    }
}
```

Anexa C – Form1.Designer.cs

```
namespace labPR4
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.tabControl1 = new System.Windows.Forms.TabControl();
            this.tabPage1 = new System.Windows.Forms.TabPage();
            this.checkBox1 = new System.Windows.Forms.CheckBox();
            this.buttonAttachments = new System.Windows.Forms.Button();
            this.button3 = new System.Windows.Forms.Button();
            this.label5 = new System.Windows.Forms.Label();
            this.txtbody = new System.Windows.Forms.TextBox();
            this.mySubject = new System.Windows.Forms.TextBox();
            this.label4 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.recieveEmail = new System.Windows.Forms.TextBox();
            this.myPassword = new System.Windows.Forms.TextBox();
            this.label2 = new System.Windows.Forms.Label();
            this.myEmail = new System.Windows.Forms.TextBox();
            this.label1 = new System.Windows.Forms.Label();
            this.tabPage2 = new System.Windows.Forms.TabPage();
            this.panel1 = new System.Windows.Forms.Panel();
            this.webBrowser1 = new System.Windows.Forms.WebBrowser();
            this.listView1 = new System.Windows.Forms.ListView();
            this.button1 = new System.Windows.Forms.Button();
            this.label8 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
            this.label7 = new System.Windows.Forms.Label();
            this.Email = new System.Windows.Forms.Label();
            this.inboxPassword = new System.Windows.Forms.TextBox();
            this.inboxEmail = new System.Windows.Forms.TextBox();
            this.tabControl1.SuspendLayout();
            this.tabPage1.SuspendLayout();
            this.tabPage2.SuspendLayout();
            this.panel1.SuspendLayout();
            this.SuspendLayout();
            //
            // tabControl1
```

```
            //
            this.tabControl1.Controls.Add(this.tabPage1);
            this.tabControl1.Controls.Add(this.tabPage2);
            this.tabControl1.Location = new System.Drawing.Point(13, 13);
            this.tabControl1.Name = "tabControl1";
            this.tabControl1.SelectedIndex = 0;
            this.tabControl1.Size = new System.Drawing.Size(968, 368);
            this.tabControl1.TabIndex = 0;
            //
            // tabPage1
            //
            this.tabPage1.Controls.Add(this.checkBox1);
            this.tabPage1.Controls.Add(this.buttonAttachments);
            this.tabPage1.Controls.Add(this.button3);
            this.tabPage1.Controls.Add(this.label5);
            this.tabPage1.Controls.Add(this.txtbody);
            this.tabPage1.Controls.Add(this.mySubject);
            this.tabPage1.Controls.Add(this.label4);
            this.tabPage1.Controls.Add(this.label3);
            this.tabPage1.Controls.Add(this.recieveEmail);
            this.tabPage1.Controls.Add(this.myPassword);
            this.tabPage1.Controls.Add(this.label2);
            this.tabPage1.Controls.Add(this.myEmail);
            this.tabPage1.Controls.Add(this.label1);
            this.tabPage1.Location = new System.Drawing.Point(4, 22);
            this.tabPage1.Name = "tabPage1";
            this.tabPage1.Padding = new System.Windows.Forms.Padding(3);
            this.tabPage1.Size = new System.Drawing.Size(960, 342);
            this.tabPage1.TabIndex = 0;
            this.tabPage1.Text = "Send Email";
            this.tabPage1.UseVisualStyleBackColor = true;
            //
            // checkBox1
            //
            this.checkBox1.AutoSize = true;
            this.checkBox1.Location = new System.Drawing.Point(888, 29);
            this.checkBox1.Name = "checkBox1";
            this.checkBox1.Size = new System.Drawing.Size(56, 17);
            this.checkBox1.TabIndex = 19;
            this.checkBox1.Text = "HTML";
            this.checkBox1.UseVisualStyleBackColor = true;
            //
            // buttonAttachments
            //
            this.buttonAttachments.Location = new System.Drawing.Point(112, 170);
            this.buttonAttachments.Name = "buttonAttachments";
            this.buttonAttachments.Size = new System.Drawing.Size(85, 42);
            this.buttonAttachments.TabIndex = 18;
            this.buttonAttachments.Text = "Attachments";
            this.buttonAttachments.UseVisualStyleBackColor = true;
            this.buttonAttachments.Click += new
System.EventHandler(this.buttonAttachments_Click);
            //
            // button3
            //
            this.button3.Location = new System.Drawing.Point(203, 170);
            this.button3.Name = "button3";
            this.button3.Size = new System.Drawing.Size(97, 42);
            this.button3.TabIndex = 16;
            this.button3.Text = "Send";
            this.button3.UseVisualStyleBackColor = true;
            this.button3.Click += new System.EventHandler(this.button3_Click);
            //
            // label5
            //
```

```
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(322, 29);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(50, 13);
this.label5.TabIndex = 15;
this.label5.Text = "Message";
//
// txtbody
//
this.txtbody.Location = new System.Drawing.Point(324, 49);
this.txtbody.Multiline = true;
this.txtbody.Name = "txtbody";
this.txtbody.Size = new System.Drawing.Size(620, 269);
this.txtbody.TabIndex = 14;
//
// mySubject
//
this.mySubject.Location = new System.Drawing.Point(112, 127);
this.mySubject.Name = "mySubject";
this.mySubject.Size = new System.Drawing.Size(188, 20);
this.mySubject.TabIndex = 7;
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(28, 130);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(43, 13);
this.label4.TabIndex = 6;
this.label4.Text = "Subject";
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(28, 104);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(47, 13);
this.label3.TabIndex = 5;
this.label3.Text = "Recieve";
this.label3.Click += new System.EventHandler(this.label3_Click);
//
// recieveEmail
//
this.recieveEmail.Location = new System.Drawing.Point(112, 101);
this.recieveEmail.Name = "recieveEmail";
this.recieveEmail.Size = new System.Drawing.Size(188, 20);
this.recieveEmail.TabIndex = 4;
//
// myPassword
//
this.myPassword.Location = new System.Drawing.Point(112, 75);
this.myPassword.Name = "myPassword";
this.myPassword.PasswordChar = '*';
this.myPassword.Size = new System.Drawing.Size(188, 20);
this.myPassword.TabIndex = 3;
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(28, 78);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(53, 13);
this.label2.TabIndex = 2;
this.label2.Text = "Password";
//
```

```
// myEmail
//
this.myEmail.Location = new System.Drawing.Point(112, 49);
this.myEmail.Name = "myEmail";
this.myEmail.Size = new System.Drawing.Size(188, 20);
this.myEmail.TabIndex = 1;
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(28, 52);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(32, 13);
this.label1.TabIndex = 0;
this.label1.Text = "Email";
//
// tabPage2
//
this.tabPage2.Controls.Add(this.panel1);
this.tabPage2.Controls.Add(this.listView1);
this.tabPage2.Controls.Add(this.button1);
this.tabPage2.Controls.Add(this.label8);
this.tabPage2.Controls.Add(this.label6);
this.tabPage2.Controls.Add(this.label7);
this.tabPage2.Controls.Add(this.Email);
this.tabPage2.Controls.Add(this.inboxPassword);
this.tabPage2.Controls.Add(this.inboxEmail);
this.tabPage2.Location = new System.Drawing.Point(4, 22);
this.tabPage2.Name = "tabPage2";
this.tabPage2.Padding = new System.Windows.Forms.Padding(3);
this.tabPage2.Size = new System.Drawing.Size(960, 342);
this.tabPage2.TabIndex = 1;
this.tabPage2.Text = "Inbox";
this.tabPage2.UseVisualStyleBackColor = true;
//
// panel1
//
this.panel1.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
this.panel1.Controls.Add(this.webBrowser1);
this.panel1.Location = new System.Drawing.Point(307, 33);
this.panel1.Name = "panel1";
this.panel1.Size = new System.Drawing.Size(637, 292);
this.panel1.TabIndex = 12;
//
// webBrowser1
//
this.webBrowser1.Dock = System.Windows.Forms.DockStyle.Fill;
this.webBrowser1.Location = new System.Drawing.Point(0, 0);
this.webBrowser1.MinimumSize = new System.Drawing.Size(20, 20);
this.webBrowser1.Name = "webBrowser1";
this.webBrowser1.Size = new System.Drawing.Size(635, 290);
this.webBrowser1.TabIndex = 11;
//
// listView1
//
this.listView1.Location = new System.Drawing.Point(19, 105);
this.listView1.Name = "listView1";
this.listView1.Size = new System.Drawing.Size(271, 220);
this.listView1.TabIndex = 10;
this.listView1.UseCompatibleStateImageBehavior = false;
//
// button1
//
this.button1.Location = new System.Drawing.Point(215, 30);
this.button1.Name = "button1";
```

```
this.button1.Size = new System.Drawing.Size(75, 23);
this.button1.TabIndex = 8;
this.button1.Text = "Login";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.button1_Click);
//
// label8
//
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(16, 80);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(33, 13);
this.label8.TabIndex = 7;
this.label8.Text = "Inbox";
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(304, 14);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(50, 13);
this.label6.TabIndex = 6;
this.label6.Text = "Message";
//
// label7
//
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(16, 59);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(53, 13);
this.label7.TabIndex = 5;
this.label7.Text = "Password";
//
// Email
//
this.Email.AutoSize = true;
this.Email.Location = new System.Drawing.Point(16, 33);
this.Email.Name = "Email";
this.Email.Size = new System.Drawing.Size(32, 13);
this.Email.TabIndex = 4;
this.Email.Text = "Email";
//
// inboxPassword
//
this.inboxPassword.Location = new System.Drawing.Point(70, 56);
this.inboxPassword.Name = "inboxPassword";
this.inboxPassword.PasswordChar = '*';
this.inboxPassword.Size = new System.Drawing.Size(139, 20);
this.inboxPassword.TabIndex = 1;
//
// inboxEmail
//
this.inboxEmail.Location = new System.Drawing.Point(70, 30);
this.inboxEmail.Name = "inboxEmail";
this.inboxEmail.Size = new System.Drawing.Size(139, 20);
this.inboxEmail.TabIndex = 0;
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(993, 393);
this.Controls.Add(this.tabControl1);
this.Name = "Form1";
this.Text = "Gmail Client";
```

```
            this.tabControl1.ResumeLayout(false);
            this.tabPage1.ResumeLayout(false);
            this.tabPage1.PerformLayout();
            this.tabPage2.ResumeLayout(false);
            this.tabPage2.PerformLayout();
            this.panel1.ResumeLayout(false);
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.TabControl tabControl1;
        private System.Windows.Forms.TabPage tabPage1;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.TextBox txtbody;
        private System.Windows.Forms.TextBox mySubject;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.TextBox recieveEmail;
        private System.Windows.Forms.TextBox myPassword;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.TextBox myEmail;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TabPage tabPage2;
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.Label Email;
        private System.Windows.Forms.TextBox inboxPassword;
        private System.Windows.Forms.TextBox inboxEmail;
        private System.Windows.Forms.Button button3;
        private System.Windows.Forms.ListView listView1;
        private System.Windows.Forms.WebBrowser webBrowser1;
        private System.Windows.Forms.Button buttonAttachments;
        private System.Windows.Forms.Panel panel1;
        private System.Windows.Forms.CheckBox checkBox1;
    }
}
```