

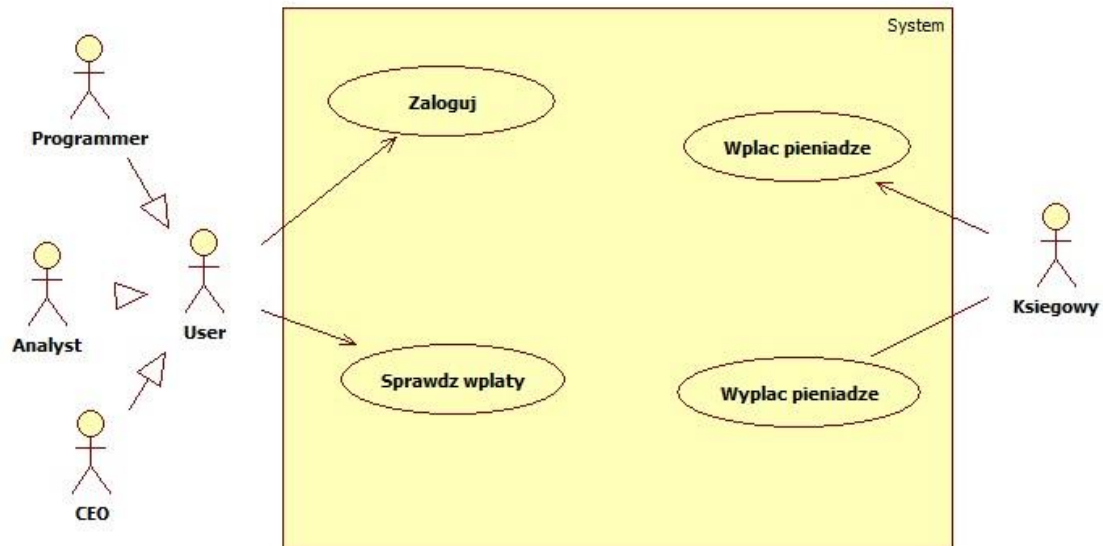
Uniwersytet Marii Curie-Skłodowskiej

System dla pracownika w firmie.

Projekt na Inżynierię Oprogramowania

Kowalczyk Łukasz
2019-06-06

Diagram przypadków użycia



Przypadki użycia

Logowanie

Przypadek użycia	Logowanie do systemu.
Scenariusz	Użytkownik loguje się do systemu na konto pracownika.
Warunki wstępne	Użytkownik posiada konto w systemie.
Niezmienniki	Użytkownik chce się zalogować na konto.
Opis	<p>Użytkownik podaje hasło i login, a następnie zatwierdza swój wybór. System weryfikuje dane.</p> <p>Scenariusz powodzenia: Użytkownik zostaje zalogowany na konto pracownika.</p> <p>Scenariusz niepowodzenia: Użytkownik dostaje komunikat o błędnych danych.</p>
Warunki końcowe	Użytkownik zostaje zalogowany na konto pracownika.

Sprawdź wpłaty

Przypadek użycia	Sprawdzenie wpłaty.
Scenariusz	Użytkownik sprawdza wpłaty, otrzymane od ostatniego sprawdzenia.
Warunki wstępne	Użytkownik jest zalogowany na konto pracownika.
Niezmienniki	Użytkownik chce sprawdzić otrzymane wpłaty.
Opis	Użytkownik wciska przycisk do sprawdzenia wpłat i otrzymuje komunikat z informacją o ilości wynagrodzenia otrzymanego od ostatniego sprawdzenia.
Warunki końcowe	Użytkownik dostaje informacje o otrzymanym wynagrodzeniu.

Wpłacić pieniądze

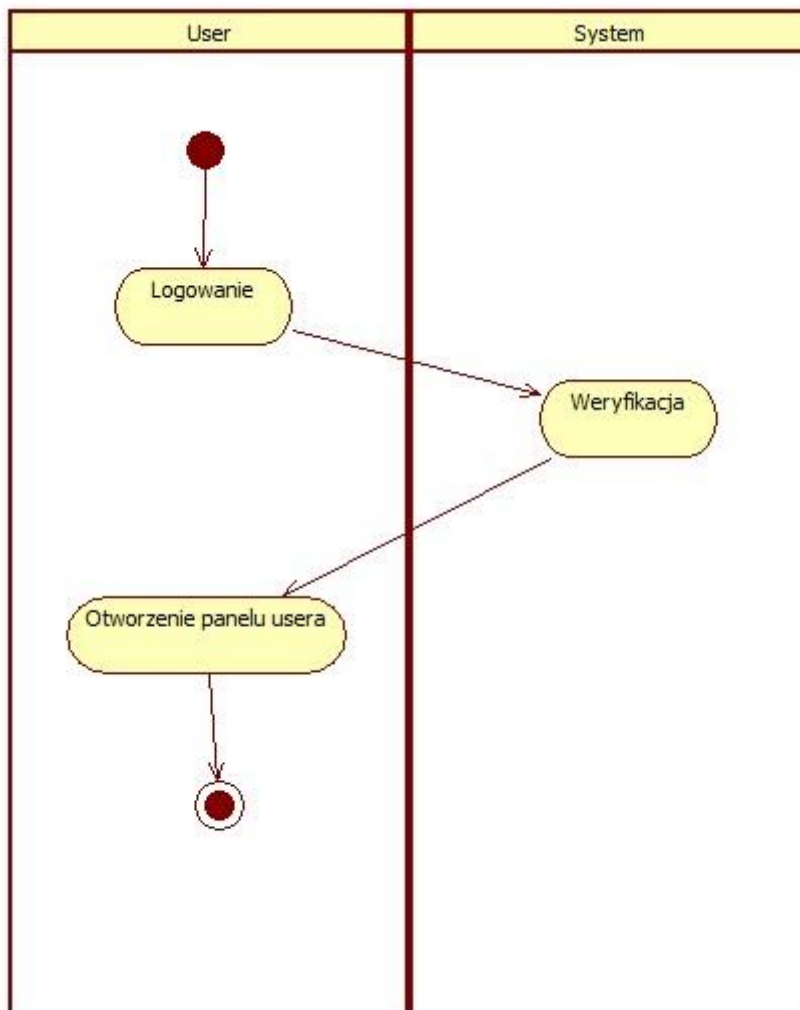
Przypadek użycia	Wpłata pieniędzy na konto bankowe.
Scenariusz	Księgowy wpłaca pieniądze na konto bankowe.
Warunki wstępne	Księgowy posiada dostęp do konta bankowego.
Niezmienniki	Księgowy dostał sygnał do wykonania wpłaty.
Opis	Księgowy wpłaca na konto firmy pieniądze.
Warunki końcowe	Stan konta ulega zmianie.

Wyplacić pieniądze

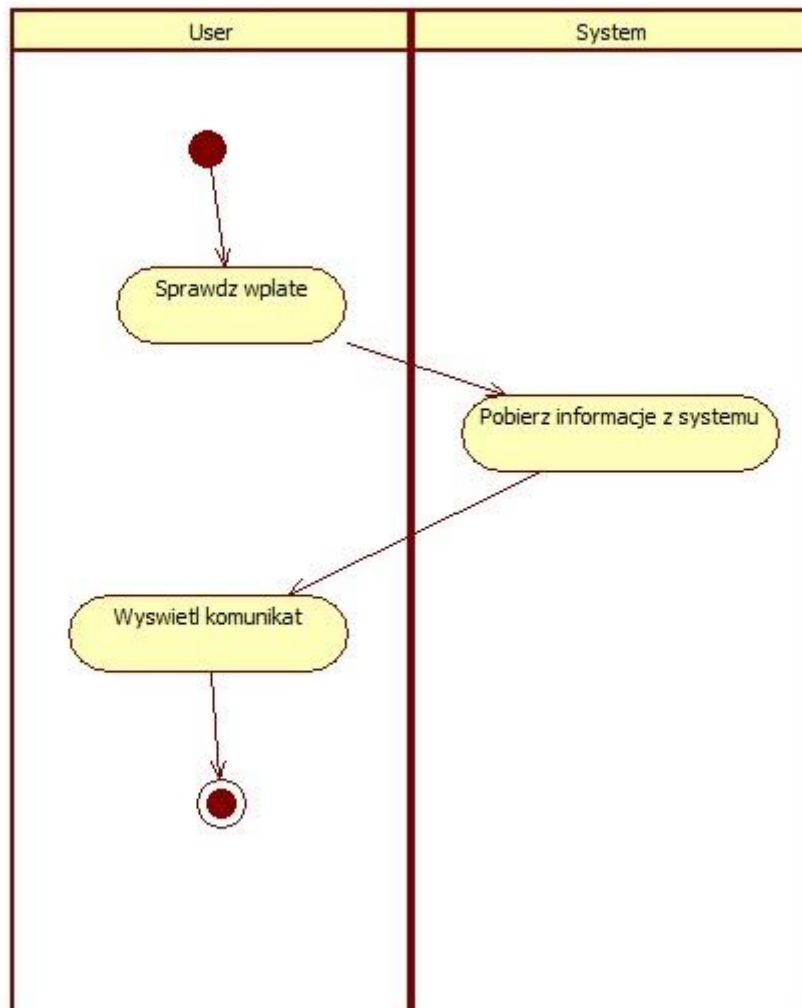
Przypadek użycia	Wyplata pieniędzy z konta bankowego.
Scenariusz	Księgowy wypłaca pieniądze z konta bankowego.
Warunki wstępne	Księgowy posiada dostęp do konta bankowego i do firmy.
Niezmienniki	Księgowy dostał sygnał do wykonania wypłaty.
Opis	<p>Księgowy odbiera od firmy informację o kwocie do pobrania. I sprawdza stan konta.</p> <p>Scenariusz powodzenia: Gdy stan konta jest większy lub równy podanej kwocie, księgowy wypłaca pieniądze z konta bankowego i wysyła firmie o wielkości pobranej kwoty.</p> <p>Scenariusz niepowodzenia: Księgowy nie pobiera pieniędzy z konta.</p>
Warunki końcowe	Stan konta ulega zmianie.

Diagramy aktywności

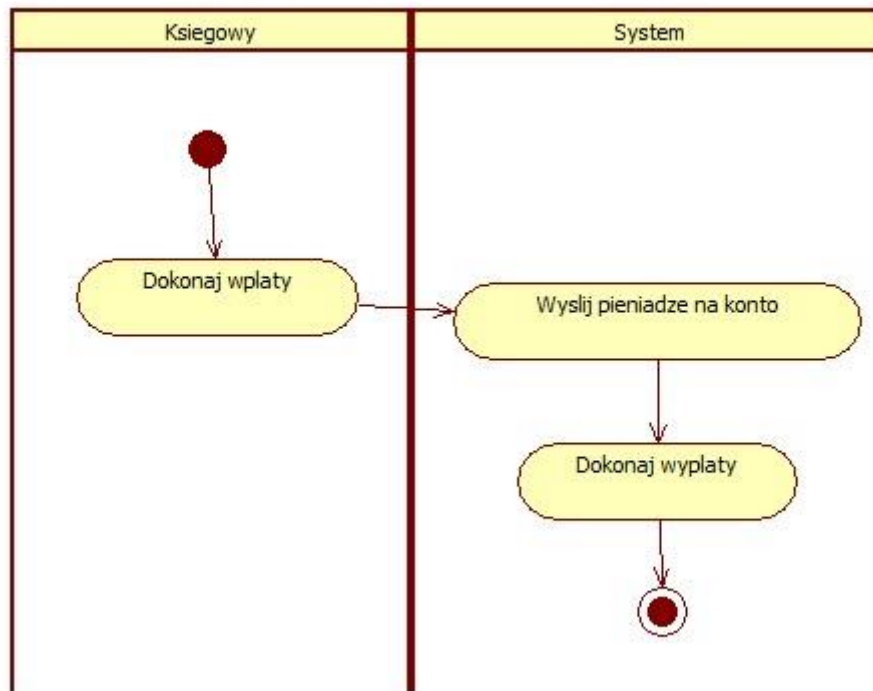
Logowanie



Sprawdź wpłatę



Wpłać pieniądze



Wypląć pieniądze

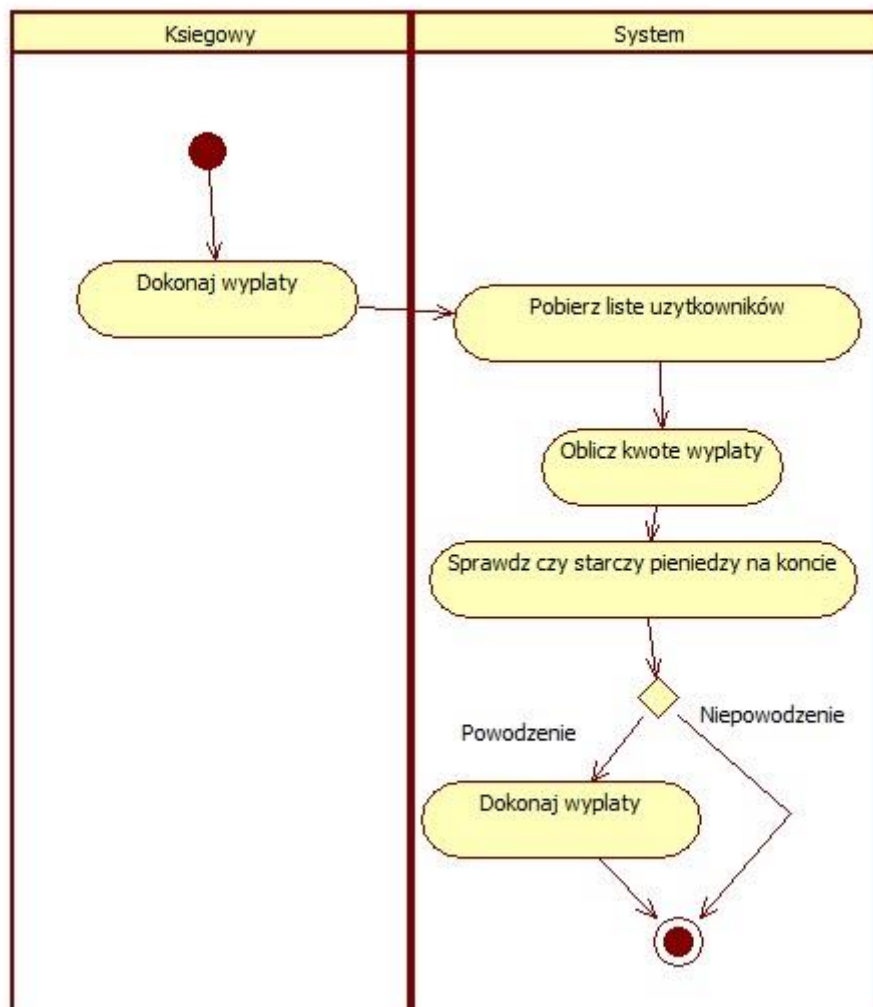


Diagram dziedziny problemu

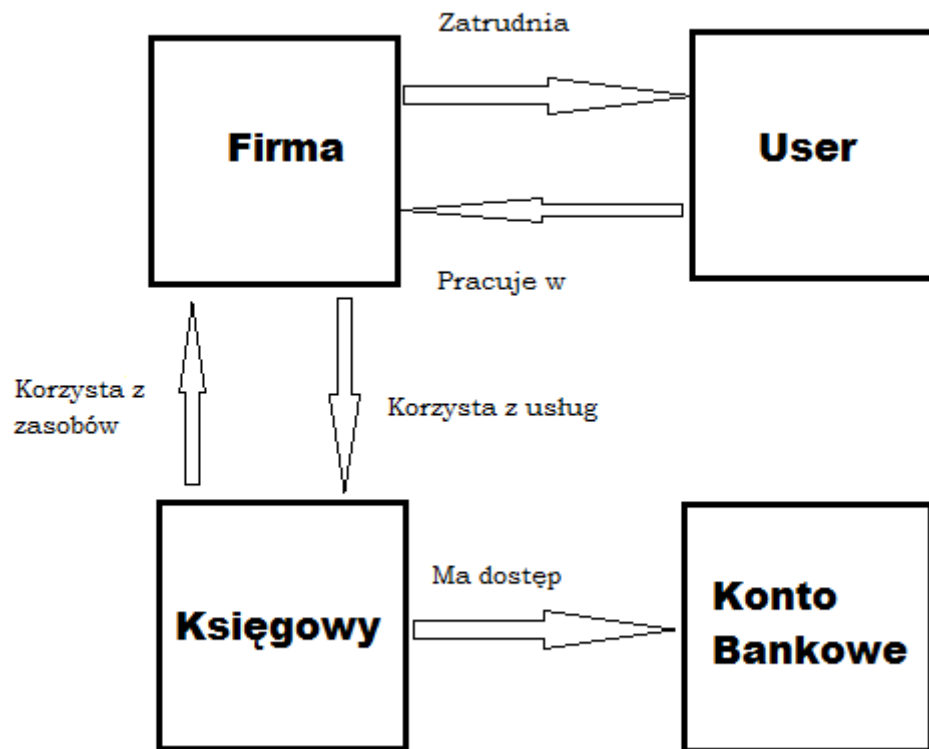


Diagram klas

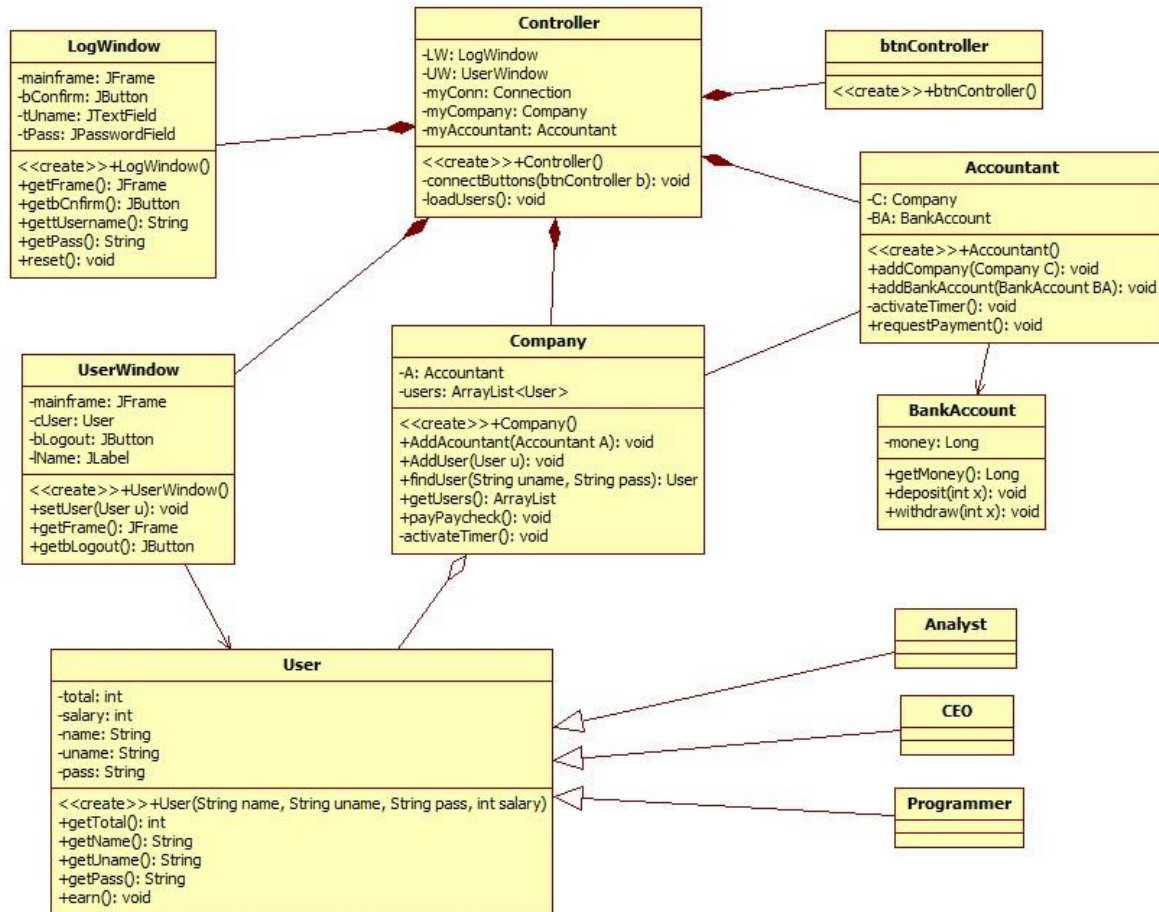
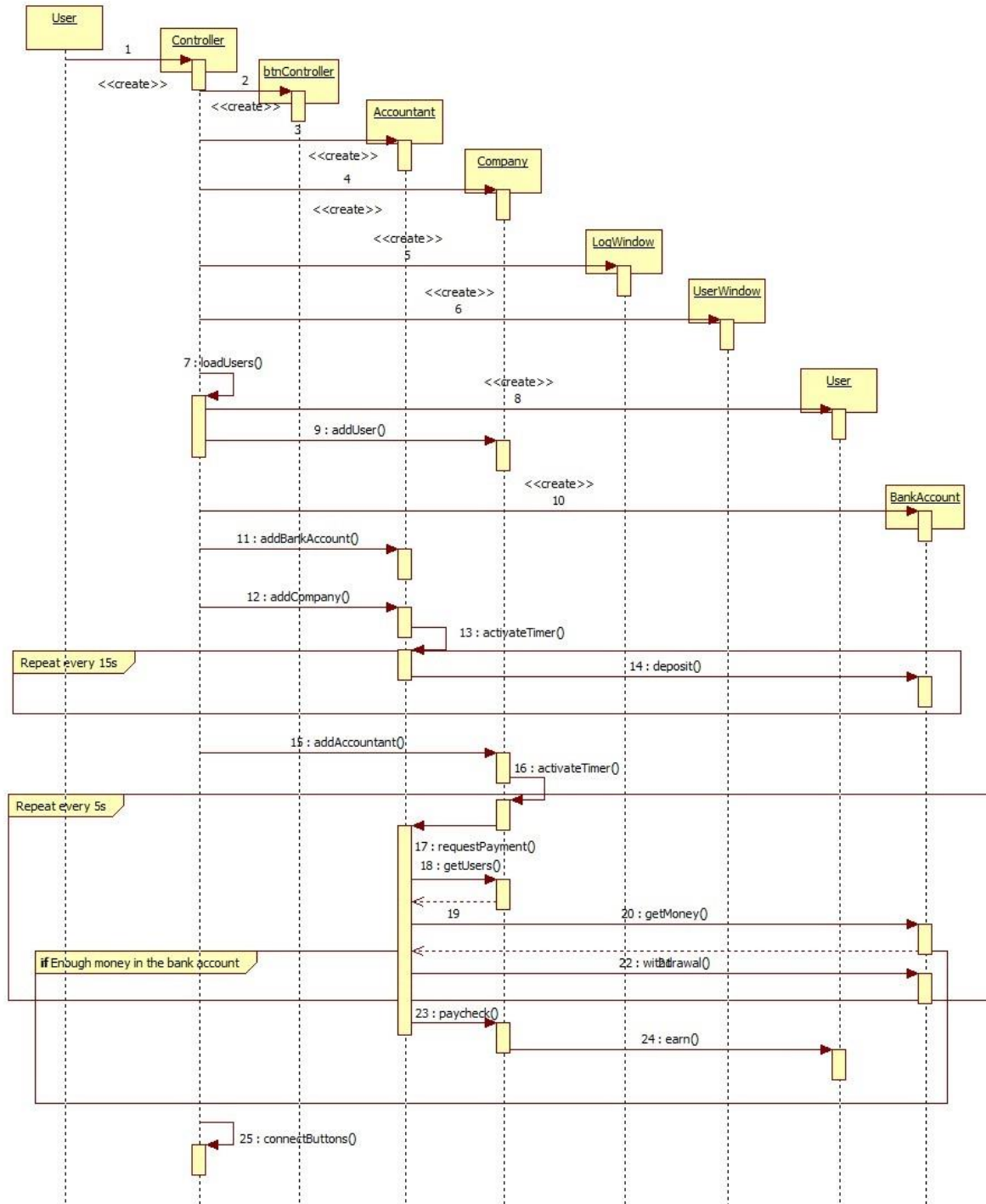
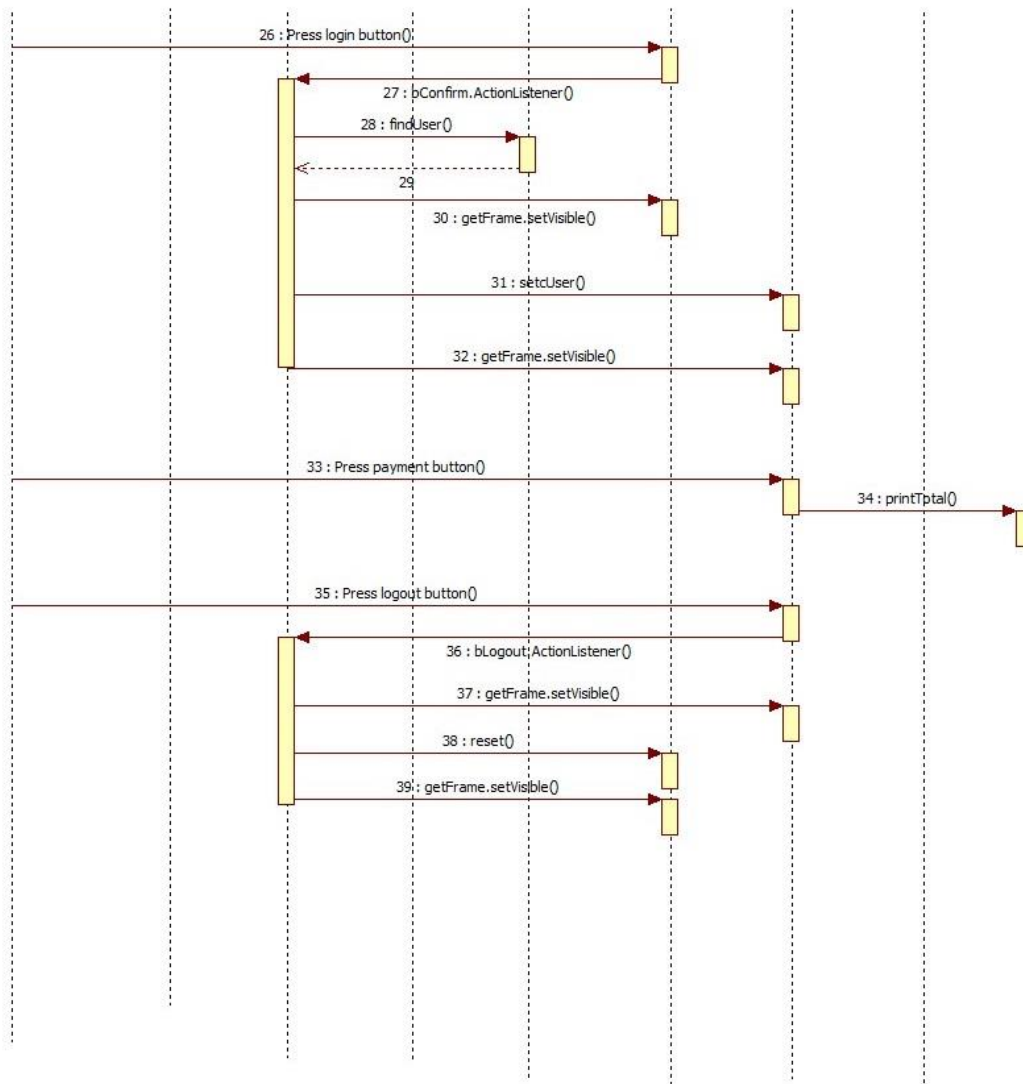


Diagram sekwencji





Implementacja

Controller.java

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class Controller {

    private LogWindow LW;
    private UserWindow UW;
    private Connection myConn;
    private Company myCompany;
    private Accountant myAccountant;

    private class btnController implements ActionListener
    {

        @Override
        public void actionPerformed(ActionEvent actionEvent) {
            Object src = actionEvent.getSource();
            if(src==LW.getbConfirm())
            {
                User u = myCompany.findUser(LW.getUsername(), LW.getPass());
                if(u!=null) {
                    LW.getFrame().setVisible(false);
                    UW.setUser(u);
                    UW.getFrame().setVisible(true);
                }
                else
                {
                    JOptionPane.showMessageDialog(null, "Niepoprawne dane",
"InfoBox: " + "Błąd", JOptionPane.INFORMATION_MESSAGE);
                }
            }
            if(src==UW.getbLogout())
            {
                UW.getFrame().setVisible(false);
                LW.reset();
                LW.getFrame().setVisible(true);
            }
        }
    }
}
```

```

    }
}

public Controller()
{
    btnController b = new btnController();
    myAccountant = new Accountant();
    myCompany = new Company();
    LW = new LogWindow();
    UW = new UserWindow();
    try {
        myConn =
DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/mydb?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC", "root", "*888rootSQL");
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
    loadUsers();
    BankAccount companyAccount = new BankAccount();
    myAccountant.addBankAccount(companyAccount);
    myAccountant.addCompany(myCompany);
    myCompany.addAccountant(myAccountant);
    connectButtons(b);
}

private void loadUsers()
{
    try
    {
        String query = "Select * from Users";
        Statement st = myConn.createStatement();
        ResultSet rs = st.executeQuery(query);
        while(rs.next()) {
            String type = rs.getString("Type");
            User u;
            if (type.equals("Programmer")) {
                u = new Programmer(rs.getString("Name"),
rs.getString("Username"), rs.getNString("Pass"));
                myCompany.addUser(u);
            }
            if (type.equals("Analyst")) {
                u = new Analyst(rs.getString("Name"),
rs.getString("Username"), rs.getNString("Pass"));
                myCompany.addUser(u);
            }
        }
    }
}

```

```

        if (type.equals("CEO")) {
            u = new CEO(rs.getString("Name"),
rs.getString("Username"), rs.getNString("Pass"));
            myCompany.addUser(u);
        }
    }
}
catch(Exception e)
{
    e.printStackTrace();
}
}

private void connectButtons(btnController b)
{
    LW.getbConfirm().addActionListener(b);
    UW.getbLogout().addActionListener(b);
}
}

```

Accountant.java

```

import java.util.Timer;
import java.util.TimerTask;

public class Accountant {

    private Company C;
    private BankAccount BA;
    public Accountant()
    {

    }

    public void addCompany(Company C)
    {
        this.C=C;
        activateTimer();
    }

    public void addBankAccount(BankAccount BA)
    {
        this.BA=BA;
    }

    public void requestPayment()
    {

```

```

        int total=0;
        for(User u: C.getUsers())
        {
            total+=u.getSalary();
        }
        if(total<=BA.getMoney())
        {
            BA.withdraw(total);
            C.payPaycheck();
        }
        System.out.println(BA.getMoney());
    }

    private void activateTimer()
    {

        Timer earnTimer = new Timer();
        earnTimer.scheduleAtFixedRate(new TimerTask() {
            @Override
            public void run() {
                BA.deposit(100000000);
            }
        },15000,15000);
    }
}

```

Company.java

```

import java.util.ArrayList;
import java.util.Timer;
import java.util.TimerTask;

public class Company {

    private Accountant A;
    private ArrayList<User> users;

    public Company()
    {
        users = new ArrayList<>();
    }

    public void addAccountant(Accountant A)
    {
        this.A=A;
        activateTimer();
    }
}

```



```

    }

    public void addUser(User u)
    {
        users.add(u);
    }

    public User findUser(String uname,String pass)
    {
        for(User u:users)
        {
            if(u.getUsername().equals(uname) && u.getPassword().equals(pass))
                return u;
        }
        return null;
    }

    public ArrayList<User> getUsers()
    {
        return users;
    }

    public void payPaycheck()
    {
        for(User u:users)
            u.earn();
    }

    private void activateTimer()
    {
        Timer timer = new Timer();
        timer.scheduleAtFixedRate(new TimerTask() {
            @Override
            public void run() {
                A.requestPayment();
            }
        }, 5000, 5000);
    }
}

```

BankAccount.java

```

public class BankAccount {

    private Long money;
    public BankAccount()
    {

```

```

        money=200000000000L;
    }

    public void withdraw(int x)
    {
        money-=x;
    }

    public void deposit(int x)
    {
        money+=x;
    }

    public Long getMoney() {
        return money;
    }
}

```

User.java

```

import javax.swing.*;

public class User {

    private String name, uname, pass;
    private int total,salary;
    public User(String name, String uname, String pass,int salary)
    {
        this.name=name;
        this.uname=uname;
        this.pass=pass;
        total=0;
        this.salary=salary;
    }

    public String getName() {
        return name;
    }

    public String getPass() {
        return pass;
    }

    public String getUname() {
        return uname;
    }
}

```

```

public void printTotal()
{
    int temp = total;
    total=0;
    JOptionPane.showMessageDialog(null, "Od ostatniego sprawdzenia
zarobiono "+temp+" PLN", "Zarobki", JOptionPane.INFORMATION_MESSAGE);
}

public void earn()
{
    total+=salary;
}

public int getSalary()
{
    return salary;
}
}

```

Programmer.java

```

public class Programmer extends User{

    public Programmer(String name, String uname, String pass)
    {
        super(name,uname,pass,12000);
    }
}

```

Analyst.java

```

public class Analyst extends User{

    public Analyst(String name, String uname, String pass)
    {
        super(name,uname,pass,15000);
    }
}

```

CEO.java

```
public class CEO extends User{

    public CEO(String name, String uname, String pass)
    {
        super(name,uname,pass,70000);
    }
}
```

LogWindow.java

```
import javax.swing.*;

public class LogWindow {

    private JFrame mainframe;
    private JButton bConfirm;
    private JTextField tUname;
    private JPasswordField tPass;

    public LogWindow()
    {
        mainframe = new JFrame("Logowanie");
        mainframe.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        mainframe.setLayout(null);
        mainframe.setResizable(false);
        mainframe.setLocationRelativeTo(null);
        int w=400,h=200;
        mainframe.setSize(w,h);

        JPanel mainpanel = new JPanel();
        mainpanel.setBounds(0,0,w,h);
        mainpanel.setLayout(null);

        bConfirm = new JButton("Zaloguj");
        bConfirm.setBounds(150,120,100,30);

        JLabel lUname = new JLabel("Login:");
        lUname.setBounds(70,30,60,30);

        JLabel lPass = new JLabel("Hasło:");
        lPass.setBounds(70,70,60,30);

        tUname = new JTextField();
        tUname.setBounds(140,30,190,30);
```

```

        tPass = new JPasswordField();
        tPass.setBounds(140, 70, 190, 30);

        mainpanel.add(bConfirm);
        mainpanel.add(lUname);
        mainpanel.add(lPass);
        mainpanel.add(tUname);
        mainpanel.add(tPass);

        mainframe.add(mainpanel);

        mainframe.setVisible(true);
    }
    public JFrame getFrame()
    {
        return mainframe;
    }

    public String getUsername()
    {
        return tUname.getText();
    }

    public String getPass()
    {
        return new String(tPass.getPassword());
    }

    public void reset()
    {
        tUname.setText("");
        tPass.setText("");
    }

    public JButton getbConfirm() {
        return bConfirm;
    }
}

```

UserWindow.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class UserWindow {

    private JFrame mainframe;
    private User cUser;
    private JLabel lName;
    private JButton bLogout;

    public UserWindow()
    {
        int w=400,h=250;
        mainframe = new JFrame("Panel pracownika");
        mainframe.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        mainframe.setLocationRelativeTo(null);
        mainframe.setLayout(null);
        mainframe.setSize(w,h);

        JPanel mainpanel = new JPanel();
        mainpanel.setLayout(null);
        mainpanel.setBounds(0,0,w,h);
        mainframe.add(mainpanel);

        lName = new JLabel("Hello");
        lName.setFont(new Font("verdana",Font.BOLD,20));
        lName.setBounds(50,20,300,40);
        lName.setHorizontalAlignment(SwingConstants.CENTER);
        mainpanel.add(lName);

        JButton bCheck = new JButton("Sprawdz wplaty");
        bCheck.setFont(new Font("Verdana",Font.BOLD,20));
        bCheck.setBounds(50,80,300,50);
        bCheck.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                cUser.printTotal();
            }
        });
        mainpanel.add(bCheck);

        bLogout = new JButton("Wyloguj");
        bLogout.setFont(new Font("Verdana",Font.BOLD,20));
        bLogout.setBounds(50,150,300,50);
```

```
        mainpanel.add(bLogout);

        mainframe.setVisible(false);
    }

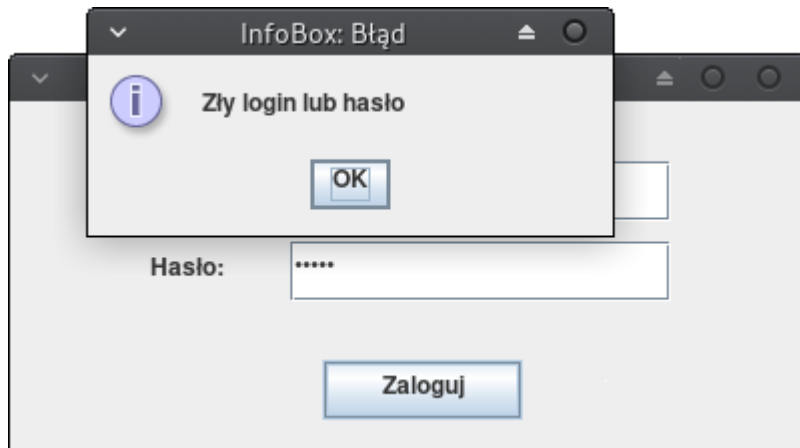
    public void setUser(User cUser) {
        this.cUser = cUser;
        lblName.setText("Witaj, "+cUser.getName()+"!");
    }

    public JFrame getFrame()
    {
        return mainframe;
    }

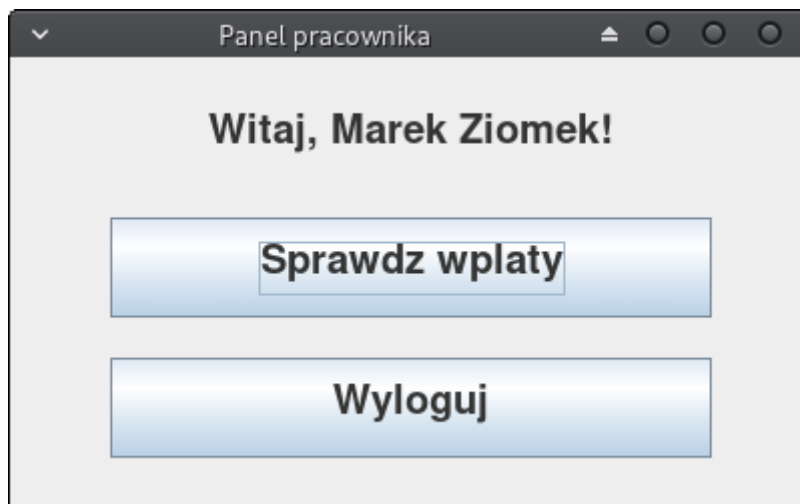
    public JButton getbLogout() {
        return bLogout;
    }
}
```

Testy

Nieudane logowanie:



Udane logowanie:



Komunikat o wynagrodzeniu:

