Dokument: Projekt aplikacji

**Projekt aplikacji**

Software Architecture Design (SAD)

# Wstęp

# Cel i zakres dokumentu

# Definicja architektury aplikacji

# Cele i ograniczenia architektury

# Obraz logiczny aplikacji

## Charakterystyka pakietów

Ze względu na niską złożoność aplikacji nie ma podziału na odrębne pakiety.

## Diagram klas aplikacji

## 

## Specyfikacja funkcji i metod aplikacji

static public void Insert(string id,string name, string species, string age,DataGrid dgAnimals)

{

string connectionString = @"Data Source=DESKTOP-9EHFHC1\ANIMALSQLSERVER;Initial Catalog=AnimalsDatabase;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionString);

con.Open();

SqlCommand cmd = new SqlCommand("insert into NewTable values (@Id,@Name,@Species,@Age)", con);

cmd.Parameters.AddWithValue("@Id", int.Parse(id));

cmd.Parameters.AddWithValue("@Name", name);

cmd.Parameters.AddWithValue("@Species", species);

cmd.Parameters.AddWithValue("@Age", int.Parse(age));

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("Insert successful");

Search(dgAnimals);

}

static public void Update(string id, string name, string species, string age, DataGrid dgAnimals)

{

string connectionString = @"Data Source=DESKTOP-9EHFHC1\ANIMALSQLSERVER;Initial Catalog=AnimalsDatabase;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionString);

con.Open();

SqlCommand cmd = new SqlCommand("Update NewTable set Name=@Name, Species=@Species, Age= @Age where ID= @ID", con);

cmd.Parameters.AddWithValue("@Id", int.Parse(id));

cmd.Parameters.AddWithValue("@Name", name);

cmd.Parameters.AddWithValue("@Species", species);

cmd.Parameters.AddWithValue("@Age", int.Parse(age));

cmd.ExecuteNonQuery();

Search(dgAnimals);

con.Close();

MessageBox.Show("Update successful");

}

static public void Delete(string id, DataGrid dgAnimals)

{

string connectionString = @"Data Source=DESKTOP-9EHFHC1\ANIMALSQLSERVER;Initial Catalog=AnimalsDatabase;Integrated Security=True";

SqlConnection con = new SqlConnection(connectionString);

con.Open();

SqlCommand cmd = new SqlCommand("Delete NewTable where ID=@ID", con);

cmd.Parameters.AddWithValue("@ID", int.Parse(id));

cmd.ExecuteNonQuery();

Search(dgAnimals);

con.Close();

MessageBox.Show("Delete successful");

}

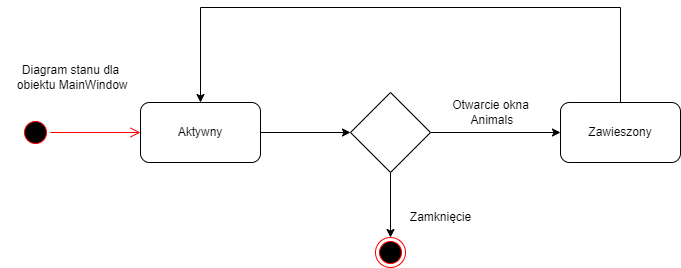
Metoda dodawania rekordu do tabeli, usuwania oraz modyfikowania wartości

# Dynamiczny obraz modelowanej aplikacji

## Diagram sekwencji UML dla obiektów

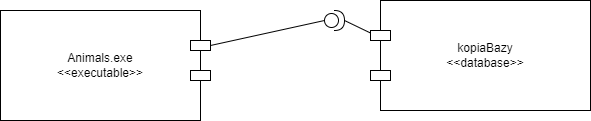
## 

## Diagram Stanu UML dla obiektów



# Statyczny obraz modelowanej aplikacji

### Diagram komponentów UML



### Diagram instalacji UML

### 

### 

# Projekt bazy danych

