

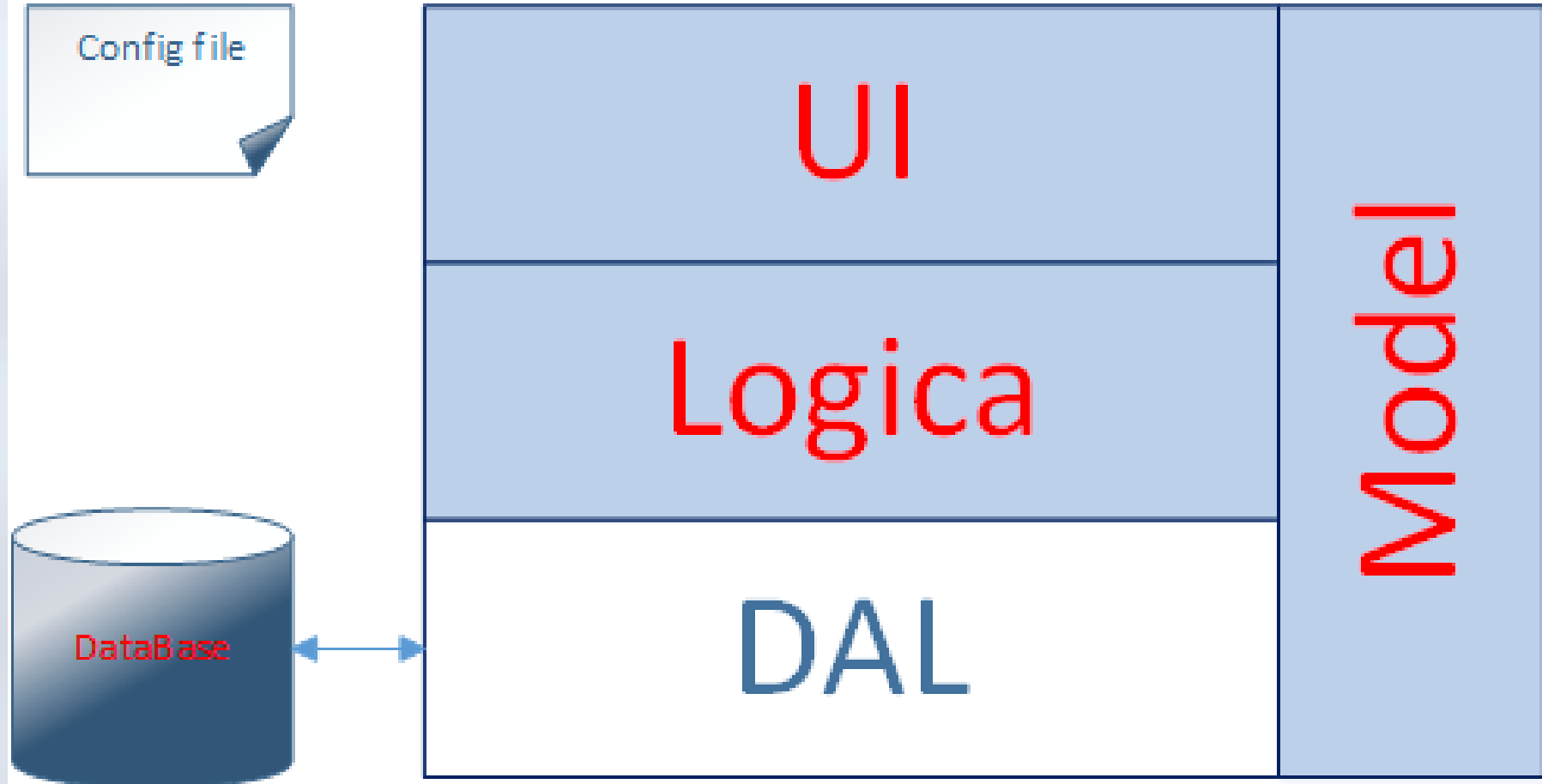


# Programming 3

# Programming 3 – week 4, 5 and 6

- Information for project Database (1.3)
  - Layered architecture
  - Model classes
  - Database access
- Information for project Application (1.4)
  - Userinterface

# Layered Architecture



# Layers: Model

- Model layer contains Model classes
  - Represent the things in the system
  - Model objects are used in all layers
  - Some examples: Customer, Book, Menu, Employee, ...

# Layers: Data Access Layer (DAL)

- contains Data Access *Objects* (DAOs)
- for each model class a DAO class
- DAL is responsible for converting the data from the database into objects, and vice versa
- Only place (layer) where SQL is used!!
- some examples: CustomerDAO, BookDAO, ReservationDAO

# Layers: User Interface Layer (UI)

- contains Windows Form (WPF, ...) classes
- responsible for making objects (partially) visible and for processing user input
- Only place where UI components are used!!
- Some examples: LoginForm, CustomersForm, SearchForm, PaymentForm

# Layers: Logic Layer

- The logic layer contains the actual system
- It is where the business logic is performed
- The logic layer contains Service classes
- In the case of simple systems, the services are organised per model class
- Some examples: CustomerService, BookService, ReservationService, ..

# This week

- Model
- DAL
- Console application for testing

## Next week

- Logic Layer
- UI Layer
- Windows Forms application

## Last week

- UI Layer: Style Guide using inheritance



# Model classes

- Classes such as Book, Customer, Programmer, Team, PlayingCard, from the assignments

# Example: Customer (Model)

```
public class Customer
{
    private int id;
    public int Id { get { return id; } set { id = value; } }

    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string EmailAddress { get; set; }

    // 'calculated' property
    public string FullName { get { return FirstName + " " + LastName; } }

    public Customer(int id, string firstName, string lastName, string emailAddress)
    {
        // ...
    }

    public override string ToString()
    {
        return $"{FullName} ({EmailAddress})";
    }
}
```

# Example: CustomerDAO

```
public class CustomerDAO
{
    private SqlConnection dbConnection;

    public CustomerDAO()
    {
        string connString = ConfigurationManager
                                .ConnectionStrings["DBConnectionString"]
                                .ConnectionString;
        dbConnection = new SqlConnection(connString);
    }

    public List<Customer> GetAll()...

    public Customer GetById(int customerId)...

    private Customer ReadCustomer(SqlDataReader reader)...
```

# Example: CustomerDAO

```
public List<Customer> GetAll()
{
    dbConnection.Open();
    SqlCommand cmd = new SqlCommand("SELECT * FROM Customers", dbConnection);
    SqlDataReader reader = cmd.ExecuteReader();
    List<Customer> customers = new List<Customer>();
    while (reader.Read())
    {
        Customer customer = ReadCustomer(reader);
        customers.Add(customer);
    }
    reader.Close();
    dbConnection.Close();

    return customers;
}
```

# Example: CustomerDAO

```
public Customer GetById(int customerId)
{
    dbConnection.Open();
    SqlCommand command = new SqlCommand(
        "SELECT * FROM Customers WHERE Id = @Id", dbConnection);

    command.Parameters.AddWithValue("@Id", customerId);

    SqlDataReader reader = command.ExecuteReader();
    Customer customer = null;
    if (reader.Read())
    {
        customer = ReadCustomer(reader);
    }
    reader.Close();
    dbConnection.Close();

    return customer;
}
```

# Example: CustomerDAO

```
private Customer ReadCustomer(SqlDataReader reader)
{
    // retrieve data from all fields
    int id = (int)reader["id"];
    string firstName = (string)reader["FirstName"];
    string lastName = (string)reader["LastName"];
    string emailAddress = (string)reader["EmailAddress"];

    // return new Customer object
    return new Customer(id, firstName, lastName, emailAddress);
}
```

# Using/testing CustomerDAO

```
void Start()
{
    CustomerDAO customerDAO = new CustomerDAO();

    // display all customers
    List<Customer> customers = customerDAO.GetAll();
    foreach (Customer cust in customers)
    {
        Console.WriteLine(cust);
    }

    // display a specific customer
    Customer customer = customerDAO.GetById(2);
    if (customer != null)
    {
        Console.WriteLine(customer);
    }
    else
    {
        Console.WriteLine("Customer not found");
    }
}
```

# Homework (for next week)

- Read paragraphs 'Yellow Book'  
*(references can be found on Blackboard)*
- Assignments week 4 *(part 1 of 'reservation system')*  
*(can be found on Blackboard)*