

## オブジェクト指向の実践 ASP.NET プログラミング

### A practice of object-oriented ASP.NET programming

ア蝕 関島淳史 松井寿夫 八月朔日元洋 所属 愛知大学蒋ゼミ  
Jiang Yong, Atsusi Sekijima, Toshio Matsui, Motohiro Hozumi

#### 1. ASP.NET による掲示板プログラミング

この掲示板システムは、オブジェクト指向プログラム言語で開発を行う。オブジェクト指向プログラムとは、プログラムを「クラス」という単位で管理し、作成したプログラムの再利用性を高めることを目的とし、「クラス」を単位とすることで、データとデータに対する処理がまとめて記述することが可能なプログラム手法である。今回はユーザーの発見→データベースデザイン→クラスの発見→実装といった開発手順とその考え方をメインに紹介する。

なぜ従来の手続き型言語を使用しないのかというと、この掲示板システムは Web Application であり多数のユーザー、不特定な場所からの使用が可能でなければならず、従来の Windows Application と比べシステムの規模は拡大し、従来のプログラミング方法で、実用的な Web Application System の開発を行うのは難しいからである。また従来の手法ではシステムの機能変更を行う場合に、変更箇所が多く、事実上変更不可能なこともあり、運用や保守また拡張性の面でも難しい。

その点、オブジェクト指向プログラム言語では、複雑かつ抽象的なシステムを分かりやすい共通言語で表現し、そのままプログラミングに実現され、開発期間を短縮するために、クラスの継承やコードの再利用という方法がとられており、従来の手法とは違い、わかりやすい開発方法、大規模なシステム開発や、その後の運用・保守・拡張性の容易さ、などを実現できる手法である。

オブジェクト指向の特徴は、対象の従属関係を発見し、対象の間に上位と下位の汎化関係を築くことであり、そのために最初にこのシステムを利用するユーザーについて考える。

##### 1.1 ユーザーの特定

まずこのシステムを開発するにあたって最初に行うことは、このシステムの利用者を特定することである。オブジェクト指向のプログラミングでは開発者側より、ユーザー側の処理からシステムの概要を考えていかななくてはならないからである。

このシステムの利用者は学生と管理者・教員だけである。学生と管理者・教員の行える処理の違いは次の図の通りである。(図 1.1)

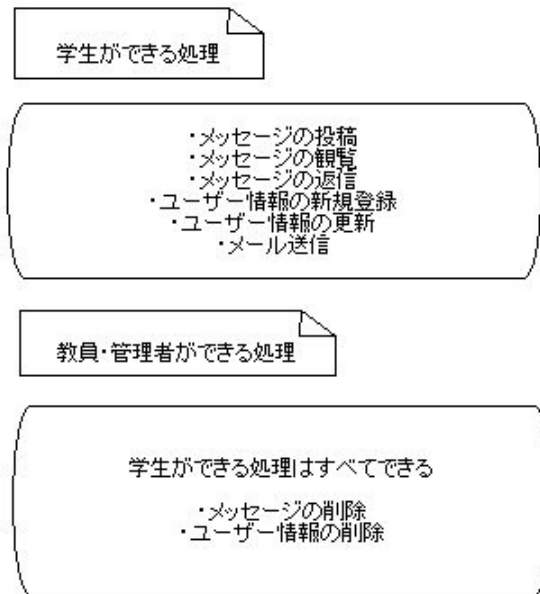


図 1.1 学生・管理者・教員の行える処理の違い

## 1.2 データベースデザイン

このようなシステムではデータベースを利用する。データベースとは『ある目的のために、関連性のあ  
る一定の情報を集めて使いやすいようにしたもの』をいう。掲示板のシステムに必要なデータをデータベ  
ース化すれば膨大なデータを最小限にまとめることができる。しかしただデータベース化しても二箇所以  
上に同じデータが繰り返すデータの冗長性を持ってしまう。このデータの冗長性はシステムにとって無駄  
な部分なので、表（テーブル）を分割するなどして正規化する。また各テーブルには id 等のように、これ  
が決まれば行を特定できるという主キーを設定する必要がある。

次にデータベースのデザインについて考える。このシステムはデータベースを利用しており、データ  
ベースにアクセスする方法には ADO.NET を使用している。ADO.NET には従来の ADO にはなかった、  
データベースを常時接続している必要がない、メモリ上にデータベースの縮小版を保持することができる、  
データベースへの問い合わせ結果をデータセットとして管理するなどの概念がとりいれられている。

またこのシステムでは Microsoft の SQLServer を利用してデータベースを構築している。

データベースのデザインについてはユーザーが行う処理の内容について考え、必要なデータをまとめて  
関係データベースを構築する。大きく 2 つに分けるとユーザーの個人情報、投稿されるメッセージに分け  
られる。

ユーザーの個人情報は、ユーザー ID、氏名、フリガナ、メールアドレス、パスワード、学生と管理者・  
教員を区別するユーザーグループ ID、プログラミングやデザイン等を区別するグループタイプ ID 等が必

要である。この中でフリガナとメールアドレス以外はシステムの構成上必ず必要なので空欄 (Null) を許さないように設定する。

投稿メッセージについてのデータには、メッセージ ID、プログラミングやデザイン等のあて先別に区別するためのメッセージタイプ ID、投稿時刻、タイトル、本文等が必要である。投稿時刻はオートで書き込まれるように設定しておく。

これらのデータの関係について、データベースを構築する方法を考えると下の図のようになる。(図 1.2)

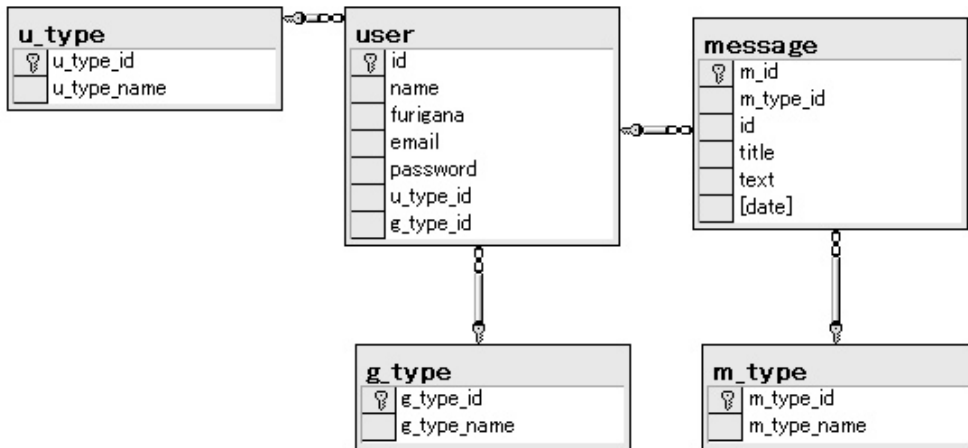


図 1.2 データベース関係図

### 1.3 クラスの定義

次にこのシステムを構築するにあたって利用するクラスについて考える。オブジェクト指向の考え方を取り入れ、クラスを利用することにした。クラスを利用することにより、多くのメリットが生まれるからだ。システムに変更する点や新しく加えたい機能が現れたとき、柔軟に対応できる。複雑なシステムでも理解しやすい。などのメリットがある。オブジェクトには属性と操作があり、それらを見つけ出すことが必要だ。目に見えないものも・抽象的なものもオブジェクトとして捉えることも必要となる。

このシステムはデータベースを使用するので、データベースに接続する、データベースに更新、登録、削除といった操作を行うデータベース専用のクラスが必要である。ほかにも先に考えたようにこのシステムの関係データベースには5つのテーブルが必要である。よってそのテーブル1つずつにデータの参照、追加を可能にするためや、プロパティの設定等のためにクラスを作成する必要がある。

またこのシステムはほとんどの機能や処理をデータベースに接続するので、データベース専用のクラスは、このシステムの表示速度に大きく関係する。表示速度を速くするためにはデータベースに接続してから更新、登録、削除を行う際に、接続開始から終了までのコードをできるだけ減らさなければならない。そのことを考えると、データベースの更新、登録、削除を行うときに必要な SQL 文は対象となるテーブルのクラスに記述し、データベースクラスではそれを受け取ることによってコードを簡略化することが有効である。

またこのシステムではメールの送信・返信を行わなければならないので、メール専用のクラスも必要である。具体的には各プロパティの設定やメールの送信機能を記述する。

よって、大まかなクラスはデータベースの処理を行うクラスとデータベースの各テーブル内のデータのプロパティの設定、対象となるテーブルの SQL 文を記述するためのクラス、そしてメールの送信についてのクラスが必要である。

利用するクラスについて図に表すと次のようになる。(図 1.3) この図では、クラス名と、その下にそのクラスのプロパティ名、そしてその下にはそのクラスのメソッド名が記述されている。

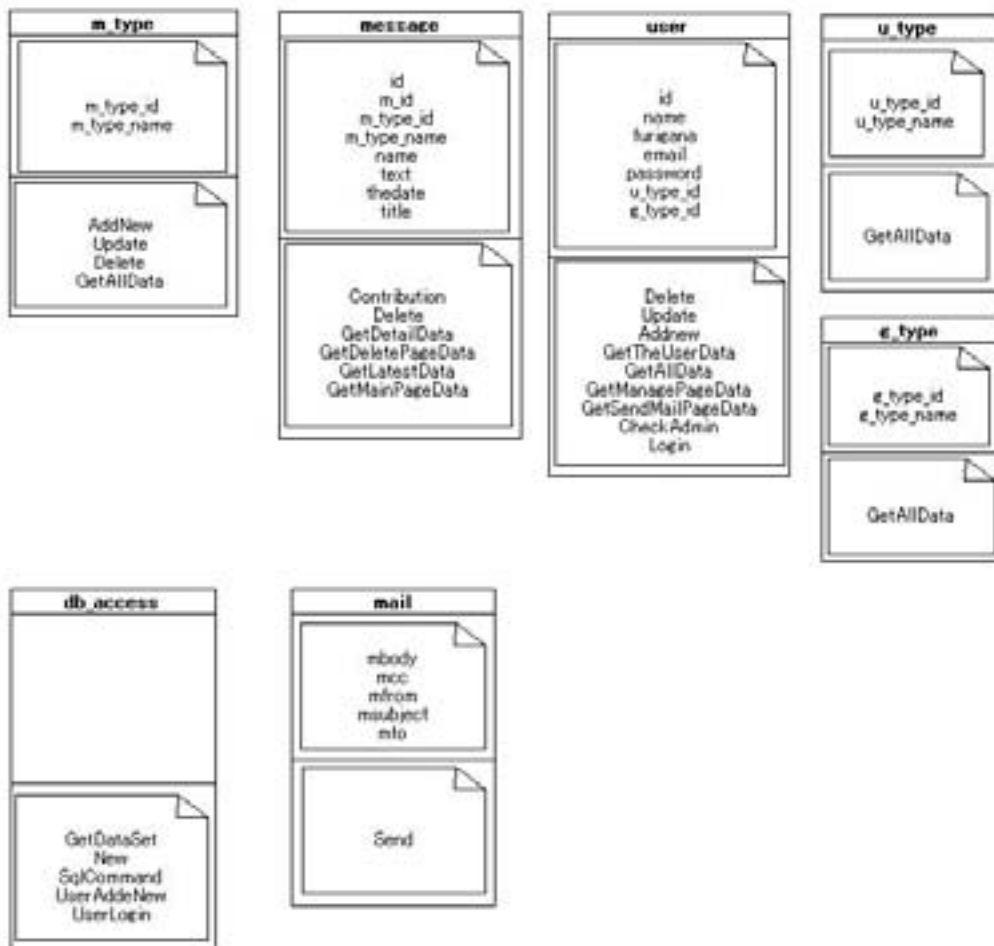


図 1.3 クラス図

## 2. ASP.NET と VB.NET の違い

.NET の Web アプリケーションは、基本的に 2 つのファイルから構成されている。

一つは、ファイル名の拡張子が.aspx となったものでその内容は HTML のタグとなっている。Web アプリケーションを実行する場合には、Web ブラウザの URL 欄に.aspx ファイルを指定する。この web のフォ

ームの配置や web 間のデータの引継ぎを Session 等を用いて容易に作成できるのが ASP.NET である。

もう一つは、ファイル名の拡張子が .vb となったもので、その内容はプログラムのコードになっている。コードは .NET Framework を通してコンパイルされ多様なプログラムを動作させる。この多様なプログラムを簡単に作成できるのが VB.NET である。

### 3. VisualStudio.NET による統合開発環境

VisualStudio.NET (図 3.1) にはコードエディタ、デザイナー、ツールボックス、サーバーエクスプローラ、プロパティウィンドウ、ソリューションエクスプローラ、クラスビュー、ダイナミックヘルプなどの機能が提供されており、これらの機能を用いて開発を行うことになる。

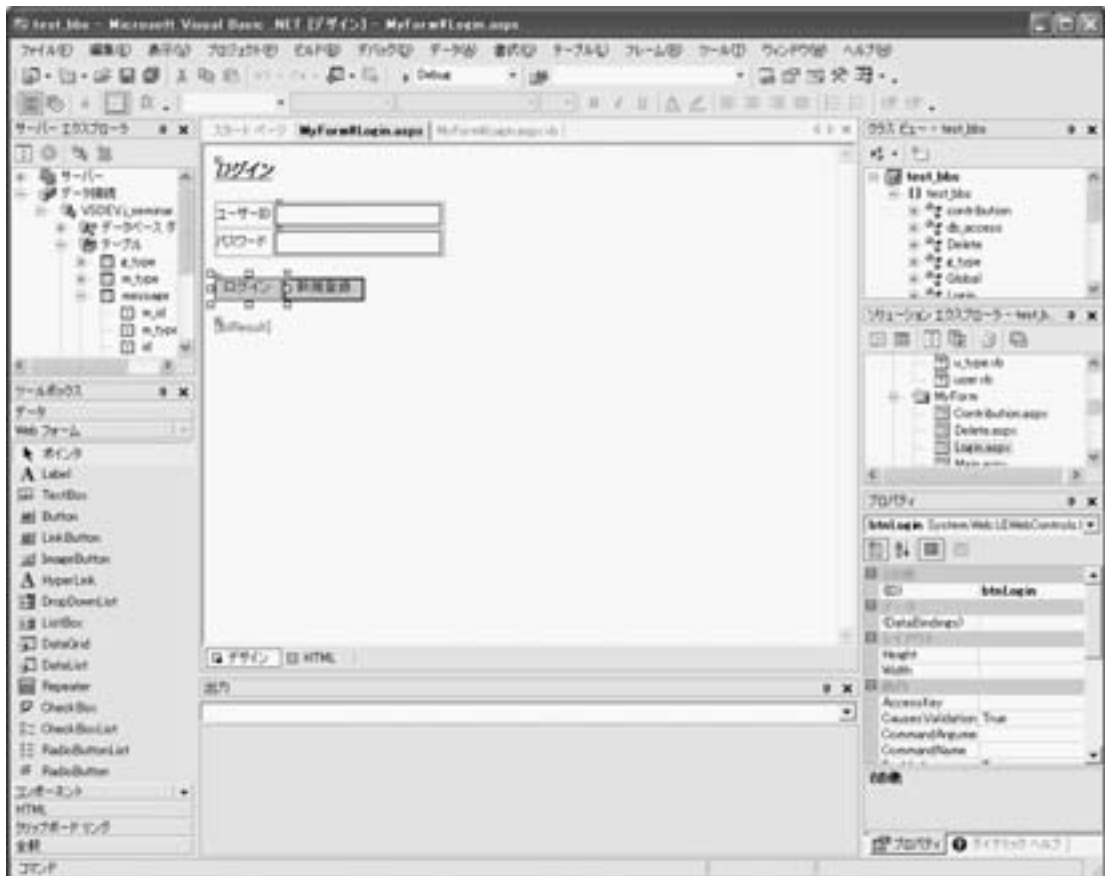


図 3.1 VisualStudio.NET 2003 の画面

コードエディタ (図 3.2) とはプログラムコードを入力するエディタで、従来通りコメントとプログラムコード部分の色分けや、自動的にインデントを追加する機能があり、デザイナーではツールボックス (図 3.3) から選択したコントロールを配置することで、フォームの作成が可能になっている。

```

スタートページ | MyForm\Login.aspx | MyForm\Login.aspx.vb
(Page Events) | Load
Public Class Login
    Inherits System.Web.UI.Page
    Protected WithEvents txtUserPassword As System.Web.UI.WebControls.TextBox
    Protected WithEvents txtUserID As System.Web.UI.WebControls.TextBox
    Protected WithEvents btnLogin As System.Web.UI.WebControls.Button
    Protected WithEvents btnAddNew As System.Web.UI.WebControls.Button
    Protected WithEvents lblTitle As System.Web.UI.WebControls.Label
    Protected WithEvents lblResult As System.Web.UI.WebControls.Label

    Web フォーム デザイナで生成されたコード

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    End Sub

    'ログイン
    Private Sub btnLogin_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLogin.Click
        '入力フィールドチェック
        If txtUserID.Text = String.Empty Then
            lblResult.Text = "ユーザーID を入力してください"
            Exit Sub
        ElseIf txtUserPassword.Text = String.Empty Then
            lblResult.Text = "パスワード を入力してください"
            Exit Sub
        End If

        'ログイン認証
        Dim aUser As New user()
        aUser.id = txtUserID.Text
        aUser.password = txtUserPassword.Text
    End Sub
End Class

```

図 3.2 コードエディタ



図 3.3 ツールボックス

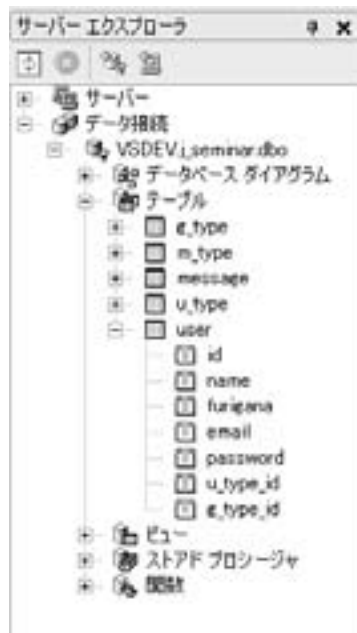


図 3.4 サーバークスプローラ



図 3.5 プロパティウィンドウ

ツールボックスには標準で多くのコントロールを利用することができ、ウィンドウも簡単で多機能なも

のが製作可能である。

サーバーエクスプローラ (図 3.4) はサーバーとサーバー上で提供されているサービスをツリー形式で表示するウィンドウであり、SQL サーバーなどのデータベースの構造を見ながら開発を行うことができ、非常に便利である。

プロパティウィンドウ (図 3.5) はプロジェクトやコントロールなどのプロパティの一覧と、その値を表示するウィンドウである。コントロールに加えて、プロパティ項目も非常に充実し、これまで難しいプログラムをかかなければならなかった処理も、プロパティ変更だけで行うことが可能である。

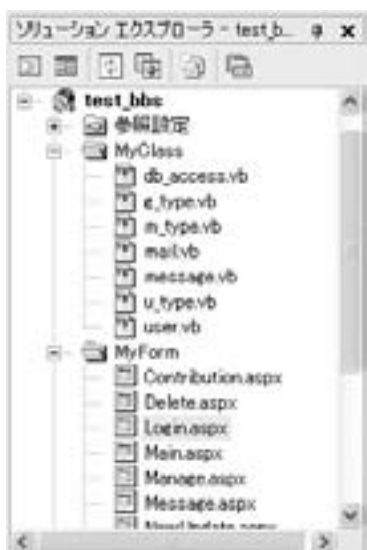


図 3.6 ソリューションエクスプローラ



図 3.7 クラスビュー

ソリューションエクスプローラ (図 3.6) はプロジェクトやプロジェクトが管理するファイル进行操作するためのウィンドウで、クラスビュー (図 3.7) はクラス間の関係やクラスの内容をツリー形式で表示するウィンドウであり、オブジェクト指向プログラミングではクラスのプロパティとメソッドなどを確認できるので重要だ。他にも、開発者の操作にあわせて、関連するドキュメントへのリンクを表示するダイナミックヘルプなどの機能もあり、開発者にとって使いやすい、利用しやすい開発環境を提供するソフトである。

## 4 実装

最後にシステムの実装について考える。つまり WEB ページのデザインとコードのプログラミングである。

ASP のプログラミングをするにはメモ帳などのテキストエディタがあれば出来る。一つのファイルのなかにコードと HTML の部分を書いていけばいいのである。今回は、ASP ではなくて、ASP.NET を選択することにした。ASP.NET は従来の ASP とは違い、コードと HTML を分離できる。そのため見やすく、開発しやすい。また、複数の言語を利用することが出来る。ASP.NET もテキストエディタで作成することが出来るが、VisualStudio.NET でコンパイルをすることで dll ファイルを作成し、その実行ファイルを動かすことでいままでより高速な処理が可能である。これが今回 ASP.NET と VisualStudio.NET を開発環境とした大きな理由である。また、その他にも .NET Framework が動作プラットフォームであるため OS に依存しない。イベントドリブン方式、XML との親和性などさまざまな特徴がある。

ページのデザインについて考えると、最初のページ (図 4.1) で利用者 (ユーザー) はログイン認証を行う必要がある。それはメインページではユーザーのタイプによって表示される項目が変わってくるため、ユーザーが学生か管理者・教員なのか管理する必要がある、また直リンクを防止するためにも必要である。ログイン認証はユーザー ID とパスワードによって行う。

そのためには最初のページからユーザー情報の新規登録や更新を行えるようにしなければならないので、ログインページにはユーザー情報の新規登録や更新を行うページに移動するためのボタンが必要である。



図 4.1 ログインページのデザイン

新規登録の追加内容は User テーブル内のすべての項目であり、更新時はユーザー ID を変更できないようにすることと、以前の登録データが表示されるようにする必要がある。

ログイン認証が完了すると、次はメインページに移動する。ASP.NET では、`Response.Redirect("index2.aspx")` とすることでページを移動できるが、移動すると前に処理した情報は引き継がれない。そのために何らかの方法で値を引き継ぐ必要がある。それには Session と Cookie と



いう方法がある。Session は、前ページで、`Session("user_name")="aichi taro"` などと値をサーバー上のリソースを利用して保存する。値を取り出すときは、`Text1.Text = Session("user_name")` などとする。この方法でログイン認証が成功したことを次のページに知らせることが出来る。Web Form の情報を簡単に次の Form へ転送できないのは、Web Application の **Statelessness** 特徴と言う。Web Application の **Staelessness** を克服するために、Cookie と Session が使われる。Web Application における Cookie と Session の使い方とセキュリティ管理は、従来の Windows Application に存在しない問題であり、十分に注意する必要がある。

メインページ (図 4.2) では投稿されたメッセージの観覧とほかのページへの移動ができなくてはならない。メッセージの観覧についてはページのデザインと見易さを考えると、メインページでは投稿されたメッセージのタイトルの一覧だけを表示して、メッセージの本文 (内容) は違うページに表示するほうが良いと思われる。また、グループタイプ別にタイトルを表示したり、新着順にタイトルを表示したりすることでもより使いやすくなる。タイトルの表示は、DataGrid を利用する。SQLServer から必要な情報を取り出し、それをハイパーリンクとしてフォームに表示させる。データベースから情報を取り出し、フォームに表示するコードは次のようなものになる。

```
Private Sub BindData()
```

```
    Dim da As SqlDataAdapter
```

```
    Dim strCon As String
```

```
    Dim ds As New DataSet
```

```
    Dim sql As String
```

```
    strCon = "Server=(local); UID=sa; PWD=*****; Database=database"
```

```
    sql = "SELECT * FROM [message]"
```

```
    da = New SqlDataAdapter(sql, strCon)
```

```
    da.Fill(ds)
```

```
    DataGrid1.DataSource = ds
```

```
    DataGrid1.DataBind()
```

```
End Sub
```

ほかのページへの移動についてはボタンで行い、メインページから移動する必要があるのは、メッセージの本文 (内容) を表示するページ、メッセージを投稿するページ、メールを送るページ、投稿されたメッセージを削除するページ、ユーザーの登録データを削除するページである。この中で、投稿されたメッセージを削除するページ、ユーザーの登録データを削除するページは管理者・教員のみにはしか使用できないので、学生のユーザーには表示されないようにしなければならない。しかし、学生用と管理者・教員用の2つのページを作るとユーザーが自分のユーザーID でログインする時に、そのID が学生か管理者・教員か判別し、管理者・教員の時はページを移動するボタンを表示し、学生の時は表示されないようにすれ

ばよい。

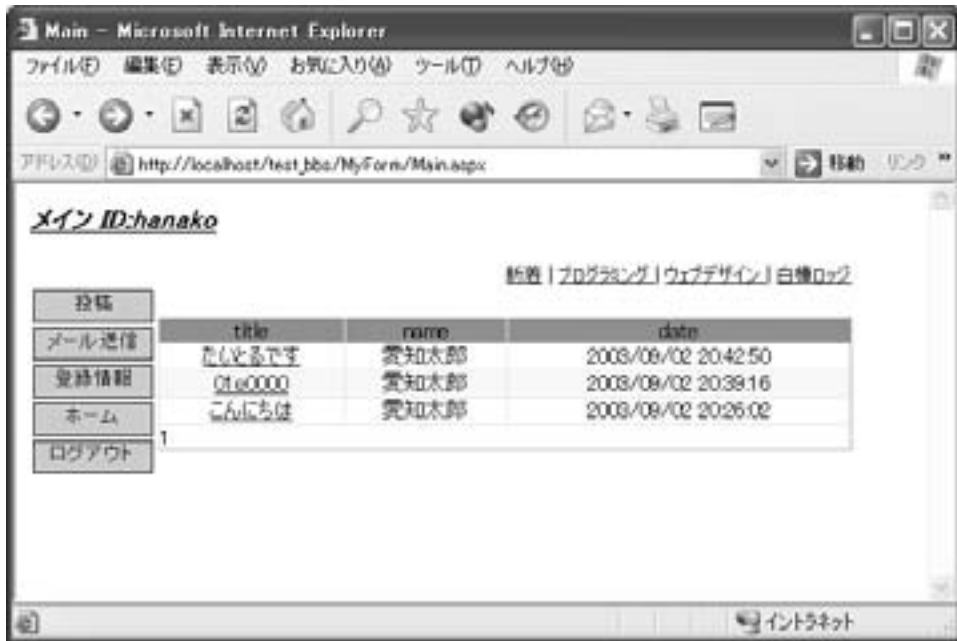


図 4.2 メインページのデザイン

メッセージを投稿するページ（図 4.3）では、メインページの機能によりグループタイプ別に分けて表示するため、投稿メッセージのグループタイプを指定する必要がある。またタイトルはメインページで表示するため空欄は許されず、絶対に必要である。



図 4.3 投稿ページのデザイン

メッセージ本文を表示するページ（図 4.4）では、本文のほかにタイトル、投稿者名、投稿者のグループ名、投稿された時刻が表示されるのが望ましい。

メールを送るページではタイトルと本文を記述でき、メールアドレスの入力ミスを防ぐために user データベースからメールアドレスの一覧をこのページに表示し、グループ名と氏名で選択する方法が最適である。



図 4.4 メッセージ送信ページのデザイン

ユーザーの登録データを削除するページ（図 4.5）では、学生ユーザーがログインした場合は表示されず、教員・管理者がログインした時のみ表示されるようにしなければならない。

投稿されたメッセージを削除するページも、学生ユーザーがログインした場合は表示されず、教員・管理者がログインした時のみ表示されるようにしなければならない。表示・非表示はオブジェクトの Enable プロパティを True/False にすることにより切り替えが出来る。

これらのページの間関係を図に表すと次のようになる。（図 4.5）

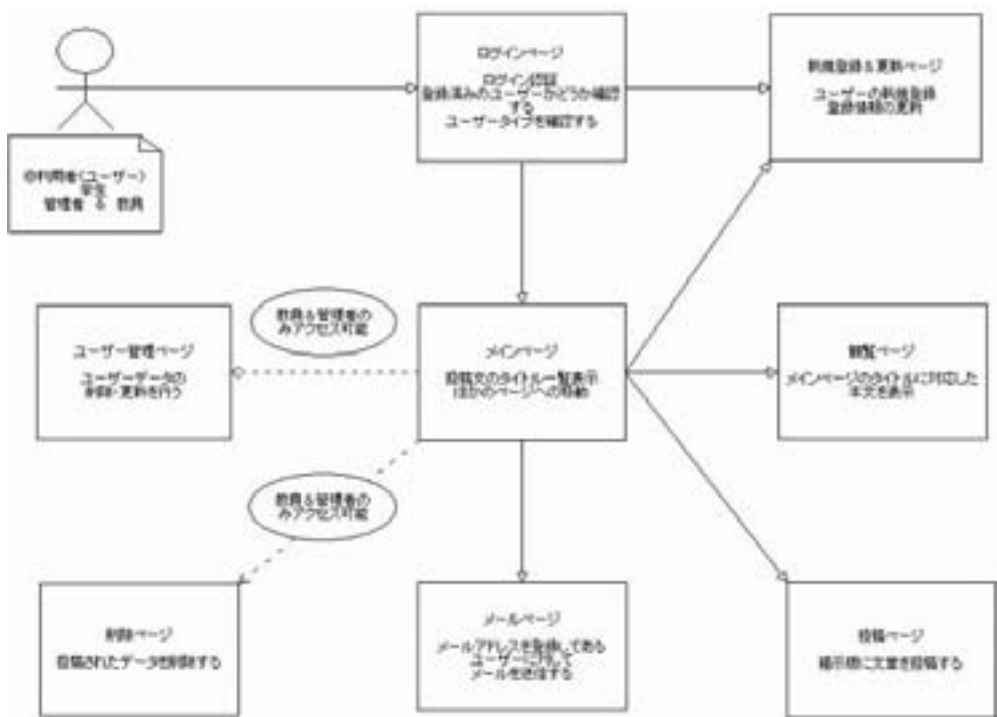


図 4.5 各ページの関係