Anhang I – Quellcodeauszüge

App_Code\lpBugreports\DataProvider.cs

DataProvider stellt Methoden bereit, die weiter unten zu sehen sind. Diese Methoden werden im SqlDataProvider gefüllt und überschrieben.

```
using System;
using DotNetNuke;
using System.Data;

using DotNetNuke.Framework;

namespace Logiplan.DNN.Modules.lpBugreports
{
  /// -----------------------------------------------------------------
  /// <summary>
  /// An abstract class that provides the DAL (Data Access Layer)
  /// </summary>
  /// -----------------------------------------------------------------
  public abstract class DataProvider
  {
    #region Shared/Static Methods

    // singleton reference to the instantiated object
    static DataProvider objProvider = null;

    // constructor
    static DataProvider()
    {
        CreateProvider();
    }

    // dynamically create provider
    private static void CreateProvider()
    {
        objProvider = (DataProvider)Reflection.CreateObject("data",
"Logiplan.DNN.Modules.lpBugreports", "");
    }

    // return the provider
    public static DataProvider Instance()
    {
        return objProvider;
    }
    #endregion

    #region Abstract Bugreports methods
    public abstract object AddBugreport(lpBugreportsInfo info);
    public abstract IDataReader GetBugreport(int bugIDInt);
    public abstract IDataReader ListBugreport();
    public abstract void UpdateBugreport(lpBugreportsInfo info);
    public abstract void DeleteBugreport(int bugIDInt);
    public abstract IDataReader GetStatusCount();
    public abstract IDataReader GetPrioritaetCount();
    #endregion

    #region Abstract BugAttachment mehtods
    public abstract void AddBugAttachment(lpBugAttachmentInfo info);
    public abstract IDataReader GetBugAttachment(int bugAttachmentID);
    public abstract IDataReader ListBugAttachment();
    public abstract void UpdateBugAttachment(lpBugAttachmentInfo info);
    public abstract void DeleteBugAttachment(int bugAttachmentID);
    #endregion
  }
}
```

App_Code\lpBugreports\SqlDataProvider.cs (Auszug)

Methoden für den Aufruf der Datenbank-Prozeduren. Es werden Daten geschrieben, oder gelesen. Alle `public override void` Methoden schreiben Daten, die anderen lesen Daten (Ausnahme `AddBugreport`, schreibt und liest).

```
#region Public Bugreports methods
public override object AddBugreport(lpBugreportsInfo info)
{
        return (object)SqlHelper.ExecuteScalar(ConnectionString,
GetFullyQualifiedName("BugreportsAdd"), info.BUGID, info.MITARBEITERID,
info.MITARBEITERIDCHANGE, info.STATUS, info.PRIORITAET, info.KUNDENID, info.PROJEKTID,
info.PRODUKTID, info.VERSIONSNUMMER, info.KURZTEXT, info.FEHLERBESCHREIBUNG,
info.MELDEDATUM, info.BEHOBENAM, info.TESTOKAM, info.DNNROLEID, info.ZUSTAENDIGKEITSID,
info.RUECKFRAGE, info.LOESUNG);
}

public override IDataReader GetBugreport(int bugIDInt)
{
      return (IDataReader)SqlHelper.ExecuteReader(ConnectionString,
GetFullyQualifiedName("BugreportsGet"), bugIDInt);
}

public override IDataReader ListBugreport()
{
      return (IDataReader)SqlHelper.ExecuteReader(ConnectionString,
GetFullyQualifiedName("BugreportsList"));
}

public override void UpdateBugreport(lpBugreportsInfo info)
{
      SqlHelper.ExecuteNonQuery(ConnectionString,
GetFullyQualifiedName("BugreportsUpdate"), info.BUGID, info.MITARBEITERID,
info.MITARBEITERIDCHANGE, info.STATUS, info.PRIORITAET, info.KUNDENID, info.PROJEKTID,
info.PRODUKTID, info.VERSIONSNUMMER, info.KURZTEXT, info.FEHLERBESCHREIBUNG,
info.MELDEDATUM, info.BEHOBENAM, info.TESTOKAM, info.DNNROLEID, info.ZUSTAENDIGKEITSID,
info.RUECKFRAGE, info.LOESUNG, info.BUGIDINT);
}

public override void DeleteBugreport(int bugIDInt)
{
      SqlHelper.ExecuteNonQuery(ConnectionString,
GetFullyQualifiedName("BugreportsDelete"), bugIDInt);
}

public override IDataReader GetStatusCount()
{
      return (IDataReader)SqlHelper.ExecuteReader(ConnectionString,
GetFullyQualifiedName("BugreportsGetStatusCount"));
}

public override IDataReader GetPrioritaetCount()
{
      return (IDataReader)SqlHelper.ExecuteReader(ConnectionString,
GetFullyQualifiedName("BugreportsGetPrioritaetCount"));
}
#endregion
```

App_Code\lpBugreports\lpBugreportsController.cs (Auszug)

Stellt die oben in der DataProvider und SqlDatraProvider bereitgestellten Methoden zur Verfügung und wandelt die Übergebenen Objekte in brauchbare Daten für die Ausgabe auf der GUI um.

```
public class lpBugreportsController
{
    #region Constructors
    public lpBugreportsController(){}
```

```
    #endregion

    #region Public Bugreports methods
    public object AddBugreport(lpBugreportsInfo info)
    {
      if (info != null)
      {
        return DataProvider.Instance().AddBugreport(info);
      }
      else
        return null;
    }

    public lpBugreportsInfo GetBugreport(int bugIDInt)
    {
      return
CBO.FillObject<lpBugreportsInfo>(DataProvider.Instance().GetBugreport(bugIDInt));
    }

    public List<lpBugreportsInfo> ListBugreport()
    {
      return
CBO.FillCollection<lpBugreportsInfo>(DataProvider.Instance().ListBugreport());
    }

    public void UpdateBugreport(lpBugreportsInfo info)
    {
      if (info != null)
      {
        DataProvider.Instance().UpdateBugreport(info);
      }
    }

    public void DeleteBugreport(int bugIDInt)
    {
      if (bugIDInt != null)
      {
        DataProvider.Instance().DeleteBugreport(bugIDInt);
      }
    }

    public List<StatusCount> GetStatusCount()
    {
      return CBO.FillCollection<StatusCount>(DataProvider.Instance().GetStatusCount());
    }

    public List<PrioritaetCount> GetPrioritaetCount()
    {
      return
CBO.FillCollection<PrioritaetCount>(DataProvider.Instance().GetPrioritaetCount());
    }
    #endregion
}
```

DesktopModules\lpBugreports\ViewlpBugreports.ascx.cs (Auszug)

Hier wird der Aufruf der Suchfunktion dargestellt. Es wird eine Controller Instanz erzeigt, die dann die Methode für das Suchen bereit stellt und Daten an das Grid auf der Übersichtsseite übergibt.

```
protected void cmdSearch_Click(object sender, EventArgs e)
{
    lpBugreportsController ctrl = new lpBugreportsController();
    List<lpBugreportsInfo> bugs = ctrl.ListBugreport();

    if (bugs.Count > 0)
    {
      grdBugs.DataSource = bugs;
      grdBugs.DataBind();
    }
}
```

DesktopModules\lpBugreports\lpBugreports.ascx (Auszug)

Hier ist der Aufbau des DataGrid auf der GUI zu sehen. Mit der Zuvor gezeigten Suchemethode werden hier die Daten mittels der DataField-Eigenschaft ans Grid gebunden. Die DataField-Eigenschaften entsprechen den Spaltennamen aus der Tabelle Bugreports.

```
<asp:DataGrid ID="grdBugs" runat="server"   AutoGenerateColumns="false"
BorderStyle="solid" BorderColor="#000000">
        <HeaderStyle CssClass="tab_head_title" HorizontalAlign="center" />
        <AlternatingItemStyle BackColor="#ECE6FE" />
        <Columns>
          ...
          <asp:BoundColumn DataField="BugIDInt" HeaderText="ID"></asp:BoundColumn>
          <asp:TemplateColumn HeaderText="Kurztext">
            <ItemTemplate>
              <%# GetKurztext(DataBinder.Eval(Container.DataItem, "Kurztext")) %>
            </ItemTemplate>
          </asp:TemplateColumn>
           ...
          <asp:TemplateColumn HeaderText="Versionsnummer">
            <ItemTemplate>
              <%# GetVersion(DataBinder.Eval(Container.DataItem, "Versionsnummer")) %>
            </ItemTemplate>
          </asp:TemplateColumn>
          <asp:TemplateColumn HeaderText="Meldedatum">
            <ItemTemplate>
              <%# GetMeldedatum(DataBinder.Eval(Container.DataItem, "Meldedatum")) %>
            </ItemTemplate>
          </asp:TemplateColumn>
           ...
        </Columns>
</asp:DataGrid>
```

DesktopModules\lpBugreports\App_LocalResources\ViewlpBugreports.ascx.de-DE.resx (Auszug)

Hier wird ein kleiner Auszug aus einer Übersetzungsdatei dargestellt, die den Aufbau der einzelnen Werte verdeutlichen soll.

```
<!-- Prioritäten -->
<data name="low.Text" xml:space="preserve">
       <value>Niedrig</value>
</data>
<data name="middle.Text" xml:space="preserve">
       <value>Mittel</value>
</data>
<data name="high.Text" xml:space="preserve">
       <value>Hoch</value>
</data>
```