Design Document for User Story 7005

As a User I want the Plan tab to have a dialog where Users can update the list of instruments for the next event.

# ASP .NET Pages

N/A

# ASP .NET User Controls

### ucPlan.ascx.cs

**Code Type:** C# Codebehind Class for ASP .NET User Control

**Uses Classes:** CCPA, CPWEvent, CDataConverter, JavaScriptSerializer, CIntake, CTemplate

**Uses PL/SQL Procedures:** N/A

**Uses MDWS Calls:** N/A

**Lines of Code:** 315

public void OnCloseInstrumentSel()

* Close the instrument selection popup

public bool InitializeControl(string strPatPWEventID)

* Initialize the control

public void LoadInstrumentGroups(long lGroupTypeID)

* load instrument groups into the combo

public void OnGroupTypeSelect(string strGroupTypeID)

* load the instrument grid based on the combo group type

public void OnGroupSelect()

* fires when a group is selected from the combo

protected void LoadInstruments()

* load instruments gridpanel by group id

public void OnInstrumentSel(string strJsonRecord)

* User checked an instrument from the instrument selection popup

public void OnInstrumentDeSel(string strJsonRecord)

* user unchecked an instrument from the instrument selection popup

public void OnSelectInstruments()

* user clicked the select instrument button

protected void Page\_Load(object sender, EventArgs e)

* page load, initialize control

# ASP .NET Classes

### CIntake.cs

**Code Type:** C# Class

**Uses Classes:** CDataParameterList, CDataSet

**Uses PL/SQL Procedures:** PCK\_INTAKE.GetModuleGroupMIDRS, PCK\_INTAKE.GetModuleGroupRS, PCK\_INTAKE.GetModuleGroupTypeRS, PCK\_INTAKE.GetPatientModulesRS, PCK\_INTAKE.GetPatientModulesTxTreeRS, PCK\_INTAKE.GetModulesListRS, PCK\_INTAKE.AssignPatientModules, PCK\_INTAKE.GetPatIntakeAssignedRS, PCK\_INTAKE.AssignInitialAssessments, PCK\_INTAKE.CompleteModule, PCK\_INTAKE.GetIntakeReport, PCK\_INTAKE.GetPatIntakeScoresRS, PCK\_INTAKE.GetIntakeReportCSVRS, PCK\_INTAKE.GetScoreDataStringRS, PCK\_INTAKE.GetNextMIDIntakeIDRS

**Uses MDWS Calls:** N/A

**Lines of Code:** 336

public DataSet GetModuleGroupMIDDS(BaseMaster BaseMstr,

long lModuleGroupID)

* get intake modules by group id

public DataSet GetModuleGroupDS(BaseMaster BaseMstr,

long lGroupTypeID)

* get module groups by group type id

# C++ Classes

N/A

# Oracle PL/SQL

### pck\_intake.bdy

**Lines of Code:** 330

procedure GetModuleGroupMIDRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nModuleGroupID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* get intake modules by group id

procedure GetModuleGroupRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nGroupTypeID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* get module groups by group type id

procedure GetModuleGroupTypeRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* get module group types such as domain and clinical setting

exception

when others then

po\_nStatusCode := 1;

po\_vStatusComment := 'PCK\_INTAKE.GetScoreDataStringRS(): ' || sqlErrm;

END;

procedure GetPatientModulesTxTreeRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

is

begin

po\_nStatusCode := 0; 0 = success

po\_vStatusComment := '';

open recordset

open rs for

select pm.,

im.module,

im.acronym,

nvl(di.complete, 0) as status,

nvl(di.intake\_id, 0) as intake\_id

from patient\_module pm,

data\_intake di,

intake\_module im

where pm.patient\_id = pi\_vPatientID

and pm.patient\_id = di.patient\_id(+)

and pm.mid = di.mid(+)

and im.mid = pm.mid(+)

and im.active = 1;

END OF LOOPS if LENGTH(v\_vDATA) > 1 then v\_vDATA := SUBSTR(v\_vDATA, 1, LENGTH(v\_vDATA) 1); end if; v\_vDATA := v\_vDATA || ']'; dbms\_output.put\_line(v\_vDATA); OPEN rs FOR SELECT v\_vDATA AS mid\_data FROM dual;

procedure GetNextMIDIntakeIDRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nMID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* Gets next module intake\_id