Design Document for User Story 7356

As a user I want an easier way to edit/update a pathway when: adding an instrument.

# ASP .NET Pages

N/A

# ASP .NET User Controls

## ucEventAddInstrument.ascx.cs

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

**Uses Classes**: CCPA, CDataConverter, CIntake

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

**Uses MDWS Calls**: N/A

**Lines of Code**: 159

public string PatPWEventID

* patient pathway event id

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

* page load

protected string GetEventMIDS()

* get comma delimeted list of instruments already added to this event

public void OnSearch()

* user searched for instruments

protected void LoadInstruments(string strGroupID)

* load instruments gridpanel by group id

public void OnGroupSelect()

* fires when a group is selected from the combo

public void OnGroupTypeSelect()

* load the instrument grid based on the combo group type

public void LoadInstrumentGroups(long lGroupTypeID)

* load instrument groups into the combo

public void Show()

* show the intrument add popup

protected void OnInstrSelOK(object sender, DirectEventArgs e)

* ok button handler for instrument add popup

# ASP .NET Classes

## CCPA.cs

**Code Type**: C# Class

**Uses Classes**: CDataParameterList, CDataSet

**Uses PL/SQL Procedures**: PCK\_INTAKE.GetModuleSearchRS, PCK\_INTAKE.GetAllModuleRS, PCK\_CPA.GetCPATypeRS, PCK\_CPA.GetCPARS, PCK\_CPA.GetCPAPWRS, PCK\_CPA.GetPWEventRS, PCK\_CPA.GetPWEventAndModuleRS, PCK\_CPA.GetPWEventModuleRS, PCK\_CPA.GetUserCPARS, PCK\_CPA.GetPatPWRS, PCK\_CPA.GetPatPWEventRS, PCK\_CPA.GetPatPWEventModuleRS, PCK\_CPA.ApplyDefaultPathway, PCK\_CPA.ApplyPathway, PCK\_CPA.UpdatePatPWEventModule, PCK\_CPA.InsertPatPWEventModules, PCK\_CPA.DeletePatPWEventModules, PCK\_CPA.UpdatePatPWEventModules

**Uses MDWS Calls**: N/A

**Lines of Code**: 539

public DataSet GetModuleSearchDS(BaseMaster BaseMstr,

string strSearch)

* get a dataset of instruments matching search criteria

public DataSet GetPWEventModuleDS(BaseMaster BaseMstr,

long lPWEventID)

* Gets all modules for a pathway event

public bool InsertPatPWEventModules(BaseMaster BaseMstr,

string strPatientID,

long lPatPWEventID,

string strMIDS,

string strFORS)

* insert instruments to the pathway event

## 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

# C++ Classes

N/A

# Oracle PL/SQL

## pck\_intake.bdy

**Lines of Code**: 389

procedure GetAllModuleRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* get rs of modules matching search criteria

procedure GetModuleSearchRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vSearch in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* get rs of modules matching search criteria

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

## pck\_cpa.bdy

**Lines of Code**: 1634

procedure UpdatePatPWEvent (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPatPWEventID in number,

pi\_vEventTitle in varchar2,

pi\_dtEventDate in date,

pi\_nAdjustFuture in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2

)

* Update the pw event info

procedure DeletePatPWEventModules (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPatPWEventID in number,

pi\_vMIDS in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2

)

* delete a scheduled module

procedure GetPWEventModuleRS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPatPWEventID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Gets all pathway event modules associated with a specific patient pathway event

procedure InsertPatPWEventModules (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPatPWEventID in number,

pi\_vMIDS in varchar2,

pi\_vModuleFors in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2

)

* insert modules into the pat pw event module list

procedure GetCPATypeRS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Gets all clinical program area types

procedure GetCPARS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nRegionID in number,

pi\_nSiteID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Gets all clinical program areas by region/site

procedure GetCPAPWRS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nCPAID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Gets all pathways for a clinical program area each site has its own list of clinical program areas

procedure GetPWEventRS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nPWID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Gets all pathway events for a pathway

procedure GetPWEventModuleRS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nPWID in number,

pi\_nPWEventID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Gets all modules for a pathway event

procedure GetPWEventAndModuleRS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nPWID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Gets all events and modules for a pathway

procedure UpdateUserDefaultCPA (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nCPAID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2

)

* update a users default clinical program area

procedure GetUserCPARS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Gets a users clinical program areas

procedure GetPatPWRS (

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

)

* Gets all pathways associated with a patient, ordered by region, site and clinical program area

procedure GetPatPWEventRS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPatPWID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Gets all pathway events for a specific patient pathway

procedure GetPatPWEventRS (

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

)

* Gets all pathway events associated with a patient

procedure GetPatPWEventRS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPatPWEventID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Gets one pathway event associated with a patient

procedure GetPatPWEventModuleRS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPatPWID in number,

pi\_nPatPWEventID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Gets all pathway event modules associated with a specific patient pathway event

procedure GetPatPWEventModuleRS (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPatPWEventID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* get all modules for a patient pw event

procedure GetPatPWEventModuleRS (

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

)

* Gets all pathway event modules associated with a patient

procedure MergePathwayEvents (

pi\_vPatientID in varchar2,

pi\_nPatPWID in number,

po\_nMerged out number

)

* merge pathway events and modules

procedure ApplyPathway (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_vPWIDs in varchar2,

pi\_vPWTitle in varchar2,

pi\_dtBaseline in date,

po\_nPatPWID out number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2

)

* apply a pathway/schedule to a patient

procedure InsertPathwayEvent (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPWID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor

)

* Insert pathway event

procedure InsertPatPWEventModules (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPatPWEventID in number,

pi\_vMIDS in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2

)

* Insert pat pw event modules

procedure UpdatePatPWEventModules (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPatPWEventID in number,

pi\_vEventTitle in varchar2,

pi\_dtEventDate in date,

pi\_nAdjustFuture in number,

pi\_vMIDS in varchar2,

pi\_vModuleFors in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2

)

* Update the pw event module list

procedure UpdatePatPWEventModule (

pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nPWID in number,

pi\_nPatPWEvtID in number,

pi\_nMID in number,

pi\_nIntakeID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2

)

* Updates Patient Event Module status after a module is completed