Design Document for User Story 6247

As a user I want the completed instruments to appear as printable reports with the option to export them as CSV files.

# ASP .NET Pages

## tbipat\_summary.aspx.cs

Code Type: C# Codebehind Class for ASP .NET Page

Uses Classes: CInstrumentResponse, StringBuilder, XslCompiledTransform, List<object>, CCPA

Uses PL/SQL Procedures: N/A

Uses MDWS Calls: N/A

Lines of Code: 177

public void ShowInstrumentReport(string strPatientID, long lMID, long lIntakeID)

* Renders and shows the completed instrument report on screen

protected void ToCsv(object sender, EventArgs e)

* Converts the instrument data to CSV format

protected void ToExcel(object sender, EventArgs e)

* Converts the instrument data to Excel spreadsheet format

protected string GetReportJSON()

* Convert/Re-format the instrument\_json session var

# ASP .NET User Controls

N/A

# ASP .NET Classes

## CInstrumentResponse.cs

Code Type: C# Class

Uses Classes: CDataParameterList, CDataSet

Uses PL/SQL Procedures: PCK\_INSTRUMENT\_RESPONSES.InsertInstrumentResponse, PCK\_INSTRUMENT\_RESPONSES.GetInstrumentResponsesRS, PCK\_INSTRUMENT\_RESPONSES.GetIntakeLogicVarRS, PCK\_INSTRUMENT\_RESPONSES.GetScoreLogicRS, PCK\_INSTRUMENT\_RESPONSES.GetPatientInstrumentsRS, PCK\_INSTRUMENT\_RESPONSES.RemovePrevResponses, PCK\_INSTRUMENT\_RESPONSES.RemoveResponse

Uses MDWS Calls: N/A

Lines of Code: 340

public DataSet GetInstrumentResponsesDS(long lMID, long lIntakeID)

* Get the instrument's responses dataset

public string GetInstrumentResponsesJson(long lMID, long lIntakeID)

* Get a JSON string of the instrument responses

public DataSet GetInstrumentResponsesDS(string strPatientID, long lMID, long lIntakeID)

* Gets and returns a dataset of responses for the completed instrument

public object GetInstrumentResponsesJSON(string strPatientID, long lMID, long lIntakeID)

* Returns a JSON object with the responses of the completed instrument

# C++ Classes

N/A

# Oracle PL/SQL

## pck\_instrument\_responses.bdy

Lines of Code: 277

procedure InsertInstrumentResponse(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nIntakeID in number,

pi\_nMID in number,

pi\_nTID in number,

pi\_nQID in number,

pi\_nRID in number,

pi\_nScoreValue in number,

pi\_vResponseValue in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* Write instrument responses to the database.

procedure GetInstrumentResponsesRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nSiteID in number,

pi\_nClinicalProgID in number,

pi\_nPathwayID in number,

pi\_nMID in number,

pi\_nIntakeID in number,

rs out PCK\_COMMON.refCursor,

po\_nStatusCode out number,

po\_vStatusComment out varchar2) is

v\_vSQL varchar2(4000);

v\_nLastIntakeID number := 0;

begin

po\_nStatusCode := 0;

po\_vStatusComment := '';

v\_vSQL := v\_vSQL || ' select dr., ';

v\_vSQL := v\_vSQL || ' im.module, ';

v\_vSQL := v\_vSQL || ' it.topic, ';

v\_vSQL := v\_vSQL || ' iq.question, ';

v\_vSQL := v\_vSQL || ' ir.response as static\_response, ';

v\_vSQL := v\_vSQL || ' ir.display\_type, ';

v\_vSQL := v\_vSQL || ' ir.unit ';

v\_vSQL := v\_vSQL || ' from data\_intake\_responses dr, ';

v\_vSQL := v\_vSQL || ' intake\_module im, ';

v\_vSQL := v\_vSQL || ' intake\_topic it, ';

v\_vSQL := v\_vSQL || ' intake\_question iq, ';

v\_vSQL := v\_vSQL || ' intake\_response ir ';

v\_vSQL := v\_vSQL || ' where dr.patient\_id = :PATIENT\_ID ';

v\_vSQL := v\_vSQL || ' and dr.mid = :MID ';

v\_vSQL := v\_vSQL || ' and dr.intake\_id = :INTAKE\_ID ';

intake\_module

v\_vSQL := v\_vSQL || ' and dr.mid = im.mid ';

intake\_topic

v\_vSQL := v\_vSQL || ' and dr.mid = it.mid ';

v\_vSQL := v\_vSQL || ' and dr.tid = it.tid ';

intake\_question

v\_vSQL := v\_vSQL || ' and dr.mid = iq.mid ';

v\_vSQL := v\_vSQL || ' and dr.tid = iq.tid ';

v\_vSQL := v\_vSQL || ' and dr.qid = iq.qid ';

intake\_response

v\_vSQL := v\_vSQL || ' and dr.mid = ir.mid ';

v\_vSQL := v\_vSQL || ' and dr.tid = ir.tid ';

v\_vSQL := v\_vSQL || ' and dr.qid = ir.qid ';

v\_vSQL := v\_vSQL || ' and dr.rid = ir.rid ';

v\_vSQL := v\_vSQL || ' order by dr.mid, ';

v\_vSQL := v\_vSQL || ' dr.tid, ';

v\_vSQL := v\_vSQL || ' dr.qid, ';

v\_vSQL := v\_vSQL || ' dr.rid ';

get lats intake\_id for the instrument

begin

select max(t.intake\_id)

into v\_nLastIntakeID

from data\_intake\_responses t

where t.patient\_id = pi\_vPatientID

and t.mid = pi\_nMID;

exception

when others then

v\_nLastIntakeID := 0;

end;

open the cursor

open rs for v\_vSQL

using pi\_vPatientID, pi\_nMID, pi\_nIntakeID;

exception

when others then

po\_nStatusCode := 1;

po\_vStatusComment := '';

end;

* Get instruments responses by MID

procedure GetScoreLogicRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nMID in number,

rs out PCK\_COMMON.refCursor,

po\_nStatusCode out number,

po\_vStatusComment out varchar2) is

v\_vSQL varchar2(4000);

begin

po\_nStatusCode := PCK\_COMMON.c\_nResultStatus\_Success;

po\_vStatusComment := '';

v\_vSQL := v\_vSQL || ' select mid, score\_logic ';

v\_vSQL := v\_vSQL || ' from intake\_module ';

v\_vSQL := v\_vSQL || ' where active = 1 ';

v\_vSQL := v\_vSQL || ' and mid = :MID ';

open the cursor

open rs for v\_vSQL

using pi\_nMID;

exception

when others then

po\_nStatusCode := PCK\_COMMON.c\_nResultStatus\_Error;

po\_vStatusComment := '';

end;

PROCEDURE GetPatientInstrumentsRS(pi\_vSessionID IN VARCHAR2,

pi\_vSessionClientIP IN VARCHAR2,

pi\_nUserID IN NUMBER,

pi\_vPatientID IN VARCHAR2,

rs OUT PCK\_COMMON.refCursor,

po\_nStatusCode OUT NUMBER,

po\_vStatusComment OUT VARCHAR2) IS

BEGIN

po\_nStatusCode := 0;

po\_vStatusComment := '';

open the cursor

OPEN rs FOR

SELECT 0 AS status\_id,

'Unknown' AS status,

t.patient\_id,

t.intake\_id,

t.mid,

m.module,

t.complete\_date AS date\_completed,

t.score,

'' AS interpretation,

DECODE(t.complete, 0, 'No', 1, 'Yes') AS completed

FROM data\_intake t, intake\_module m

WHERE t.mid = m.mid

AND t.patient\_id = pi\_vPatientID;

EXCEPTION

WHEN OTHERS THEN

po\_nStatusCode := 1;

po\_vStatusComment := '';

END;

* Get the score logic for the module

procedure RemovePrevResponses(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nMID in number,

pi\_nIntakeID in number,

pi\_vResponses in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* Remove responses from previous time submitted not present in a new submittal of the instrument