Design Document for User Story 7242

As a user I will like to specify the Preferred Language of a patient when this is not available in the predefine list of the demographics section.

# ASP .NET Pages

### pat\_demographics.aspx.cs

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

**Uses Classes**: CStat, CPatientDemographics, List<string>

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

**Uses MDWS Calls**: N/A

**Lines of Code**: 381

public void UpdatePatientDemographics()

* Collects the page data and submits the calls to the stored procedures that update patient demographics and socioeconomics

protected bool ValidateDemographicsData(out string strMessage)

* Validates the data entered in the demographics tab

# ASP .NET User Controls

N/A

# ASP .NET Classes

### CPatientDemographics.cs

**Code Type**: C# Class

**Uses Classes**: CDataParameterList, CDataSet

**Uses PL/SQL Procedures**: PCK\_PATIENT.GetPatientDemographicsRS, PCK\_PATIENT.GetPatientSocioeconomicsRS, PCK\_PAT\_ETHNICITY\_RACE.GetPatientRaceRS, PCK\_PAT\_ETHNICITY\_RACE.GetPatientEthnicityRS, PCK\_PATIENT.UpdatePatientDemographics, PCK\_PATIENT.UpdatePatientSocioeconomics, PCK\_PAT\_ETHNICITY\_RACE.UpdatePatientRace, PCK\_PAT\_ETHNICITY\_RACE.UpdatePatientEthnicity

**Uses MDWS Calls**: N/A

**Lines of Code**: 221

public bool UpdatePatientDemographics(string strFName,

string strMi,

string strLName,

string strSponsorSSN,

string strSSN,

string strGender,

string strDOB,

string strProviderID,

string strAddress1,

string strAddress2,

string strCity,

string strPostalCode,

string strHomePhone,

string strCellPhone,

string strWorkPhone,

string strEmail,

string strStateID,

long lDutyStatusID,

long lPrefLangID,

string strPrefLangOther,

long lHandPrefID,

long lCallPreference,

long lHomePhoneMsg,

long lEmailMsg)

* Updates patient demographics record with the patient demographics page data

# C++ Classes

N/A

# Oracle PL/SQL

### pck\_patient.bdy

**Lines of Code**: 981

procedure UpdatePatientDemographics(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

pi\_vPatientID in varchar2,

pi\_vFirstName in varchar2,

pi\_vMI in varchar2,

pi\_vLastName in varchar2,

pi\_vSponsorSSN in varchar2,

pi\_vSSN in varchar2,

pi\_vGender in varchar2,

pi\_vDateOfBirth in varchar2,

pi\_vProviderID in varchar2,

pi\_vAddress1 in varchar2,

pi\_vAddress2 in varchar2,

pi\_vCity in varchar2,

pi\_vPostal\_Code in varchar2,

pi\_vHomePhone in varchar2,

pi\_vCellPhone in varchar2,

pi\_vWorkPhone in varchar2,

pi\_vEmail in varchar2,

pi\_vStateID in varchar2,

pi\_nCellPhoneMsg in number,

pi\_nEmailMsg in number,

pi\_nCallPreference in number,

pi\_nPrefLangID in number,

pi\_vPrefLangOther in varchar2,

pi\_nDutyStatusID in number,

pi\_nHandPrefID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* Updates patient demographics data from the demographics page in the application

procedure GetPatientLookupRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

pi\_nSelectedCases in number,

pi\_nSearchType in number,

pi\_vSearchValue in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* get a record set of patients back that match criteria

procedure GetPatientDemographicsRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

pi\_vPatientID in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* Returns the patient demographics recordset

duty\_status\_id,

hand\_preference\_id

from tbicds.patient\_demographics t

where t.patient\_id = pi\_vPatientID;

exception

when others then

po\_nStatusCode := 1;

po\_vStatusComment := 'PCK\_PATIENT.GetPatientDemographicsRS(): ' ||

sqlErrm;

end;

converted to use new encryption

procedure InsertPatientDemographics(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

Patient Table

pi\_vKey in varchar2,

pi\_vPatientID in varchar2,

pi\_vEncounterID in varchar2,

pi\_vFirstName in varchar2,

pi\_vMI in varchar2,

pi\_vLastName in varchar2,

pi\_vSponsorSSN in varchar2,

pi\_vSSN in varchar2,

pi\_vGender in varchar2,

pi\_vDateOfBirth in varchar2,

pi\_vProviderID in varchar2,

pi\_vAddress1 in varchar2,

pi\_vAddress2 in varchar2,

pi\_vCity in varchar2,

pi\_vPostal\_Code in varchar2,

pi\_vHomePhone in varchar2,

pi\_vCellPhone in varchar2,

pi\_vWorkPhone in varchar2,

pi\_vEmail in varchar2,

pi\_vStateID in varchar2,

pi\_nCellPhoneMsg in number,

pi\_nEmailMsg in number,

pi\_nCallPreference in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

tbicds.fnc\_utl\_decstr(t.preferred\_language\_other, pi\_vKey, t.PATIENT\_ID) as preferred\_language\_other,

insert into tbicds.treatment

(treatment\_id,

patient\_id,

initial\_visit\_date,

last\_updated,

last\_updated\_by)

values

(1, pi\_vPatientID, sysdate, sysdate, pi\_nUserID);

commit;

REVAMP 20130219 DS: As requested by VA (Dr Kuna), do not create a default encounter

insert into tbicds.encounter

(encounter\_id,

treatment\_id,

patient\_id,

encounter\_date,

last\_updated,

last\_updated\_by)

values

(pi\_vEncounterID, 1, pi\_vPatientID, sysdate, sysdate, pi\_nUserID);

commit;

exception

when others then

po\_nStatusCode := 1;

po\_vStatusComment := 'PCK\_PATIENT.InsertPatientDemographics(): ' ||

sqlErrm;

end;

does the patient only have 1 encounter

for the latest treatment?

procedure InitialEncounter(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

po\_nInitialEnc out number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

insert into tbicds.patient\_emergency\_contact (patient\_id, relationship\_id, last\_updated, last\_updated\_by) values (pi\_vPatientID, v\_nRelationshipID, sysdate, pi\_nUserID); commit;

exception

when others then

po\_nStatusCode := 1;

po\_vStatusComment := 'PCK\_PATIENT.DelIncPatIntakeAssessments(): ' ||

sqlErrm;

end;

procedure IncPatIntakeAssessments(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

po\_nHasIncPatAssessments out number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

for rec in ( select all INCOMPLETE encounter intakes for module for patient select ei. from tbicds.encounter\_intake ei where ei.encounter\_id in ( select all encounters for patient select e.encounter\_id from tbicds.encounter e where e.patient\_id = pi\_vPatientID)) loop begin v\_nNumOfIncAssessments := 0; select all encounter intakes responses for module for encounter by mid select count(eir.encounter\_id) into v\_nNumOfIncAssessments from tbicds.encounter\_intake\_responses eir where eir.encounter\_id = rec.encounter\_id and eir.mid = rec.mid; if v\_nNumOfIncAssessments = 0 then delete from tbicds.encounter\_intake ei where ei.encounter\_id = rec.encounter\_id and ei.encounter\_intake\_id = rec.encounter\_intake\_id and ei.mid = rec.mid; commit; end if; exception when others then rollback; end; end loop;

exception

when others then

po\_nStatusCode := 1;

po\_vStatusComment := 'PCK\_PATIENT.IncPatIntakeAssessments(): ' ||

sqlErrm;

end;

procedure DeletePatientAssignedModule(pi\_vPatientID in varchar2,

nMID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

for rec in ( select all INCOMPLETE encounter intakes for module for patient select ei. from tbicds.encounter\_intake ei where ei.encounter\_id in ( select all encounters for patient select e.encounter\_id from tbicds.encounter e where e.patient\_id = pi\_vPatientID)) loop begin v\_nNumOfIncAssessments := 0; select all encounter intakes responses for module for encounter by mid select count(eir.encounter\_id) into v\_nNumOfIncAssessments from tbicds.encounter\_intake\_responses eir where eir.encounter\_id = rec.encounter\_id and eir.mid = rec.mid; if v\_nNumOfIncAssessments = 0 then po\_nHasIncPatAssessments := 1; end if; exception when others then rollback; end; end loop;

procedure GetPatientSocioeconomicsRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

pi\_vPatientID in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* Returns patient socioeconomics recordset

procedure UpdatePatientSocioeconomics(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

pi\_vPatientID in varchar2,

pi\_nHighestGradeID in number,

pi\_nHighestDegreeID in number,

pi\_nMaritalStatusID in number,

pi\_nEmploymentStatusID in number,

pi\_nJobClassificationID in number,

pi\_vEmploymentStatusOther in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* Updates patient socioeconomics data

procedure InsertPatientSocioeconomics(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

pi\_vPatientID in varchar2,

pi\_nHighestGradeID in number,

pi\_nHighestDegreeID in number,

pi\_nMaritalStatusID in number,

pi\_nEmploymentStatusID in number,

pi\_nJobClassificationID in number,

pi\_vEmploymentStatusOther in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* Inserts patient socioeconomics data

### pck\_pat\_ethnicity\_race.bdy

**Lines of Code**: 377

procedure GetPatientEthnicityRS(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)

* Returns patient ethnicity recordset

procedure GetPatientRaceRS(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)

* Returns patient race recordset

procedure UpdatePatientRace(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_vValues in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* Updates patient race selected values

procedure UpdatePatientEthnicity(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPatientID in varchar2,

pi\_nEthnicityID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* Updates patient ethnicity