# Design Document for User Story 5716

## As a User I want the TBI-CDS application to lookup the patient from CPRS automatically.

## ASP .NET Pages

### TBICDSX.aspx.cs

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

Uses Classes: CMDWSOps, CSec

Uses PL/SQL Procedures: N/A

Uses MDWS Calls: N/A

Lines of Code: 69

* protected void Page\_Load(object sender, EventArgs e)
  + logs in and looks up the patient. used by the thin cleint to access the website.

### TBICDSXPat.aspx.cs

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

Uses Classes: CMDWSOps

Uses PL/SQL Procedures: N/A

Uses MDWS Calls: N/A

Lines of Code: 36

* protected void Page\_Load(object sender, EventArgs e)
  + check that user is logged in, sanitize querystring values and automatically lookup the patient

## ASP .NET User Controls

N/A

## ASP .NET Classes

### MasterPage.master.cs

Code Type: C# Class

Uses Classes: CUser, CDataUtils, CTreatment, DataSet, CSystemSettings, CPatient, CAppMenu, CUserAdmin, CPatientLock, CMessages, CEncounter

Uses PL/SQL Procedures: N/A

Uses MDWS Calls: N/A

Lines of Code: 293

* public void LookupPatient(string strPatientID)
  + US:5729 US:5439 public method to lookup a patient by patient id

### BaseMaster.cs

Code Type: C# Class

Uses Classes: EmrSvcSoapClient, CData, CPatientLock, CSec, CDataConnection, AppMaster, CDataParameterList, Random, CUser, CDataUtils

Uses PL/SQL Procedures: PCK\_FX\_SEC.SetSessionValue, PCK\_FX\_SEC.GetSessionValue, PCK\_FX\_SEC.DeleteSessionValue, PCK\_FX\_SEC.DeleteAllSessionValues

Uses MDWS Calls: N/A

Lines of Code: 485

* public void LookupPatientByDUZ(string strDUZ, string strDFN)
  + patient lookup by duz
* public void GotoHomePage()
  + redirects to our home page
* public void LogOff()
  + called to logoff the user

### CMDWSOps.cs

Code Type: C# Class

Uses Classes: CStatus, CMDWSTransfer, EmrSvcSoapClient, CAppUser

Uses PL/SQL Procedures: N/A

Uses MDWS Calls: select, match, getUserSecurityKeys, getVHA, getNoteTitles, getTeams, getSpecialties, getWards, getClinics, getPatientsByTeam, getPatientsBySpecialty, getPatientsByWard, getPatientsByClinic, cprsUserLookup, getPatientsByProvider, connect, login, isValidEsig, writeNote, signNote, closeNote, getNotesWithText, getLabTests

Lines of Code: 492

* public CStatus TransferPatientByID(string strkey,

string strDFN,

out string strPatID)

* + transfer a patient by DFN

### CMDWSTransfer.cs

Code Type: C# Class

Uses Classes: CPatientDataItemXfer, CPatientDataXfer, CStatus, EmrSvcSoapClient, CDataParameterList, CDataSet, CSiteDataXfer, CPatientItemDataItem, CPatientItemComponentDataItem, CItemComponentData, CPatientItemCompList, CPatientChecklistLogic, CLabDataXfer, CTeamDataXfer, CNoteTitleDataXfer, CSpecialtyDataXfer, CWardDataXfer, CSecurityKeyDataXfer, CClinicDataXfer, CUserDataXfer

Uses PL/SQL Procedures: PCK\_MDWS.GetMDWSPatientRS

Uses MDWS Calls: select, getUnsignedNotes, getNotesWithText, getChemHemReports, getLabTestDescription

Lines of Code: 822

* public CStatus TransferPatient(HttpSessionState WebSession,

string strKey,

PatientTO toPatient,

out string strPatID)

* + Transfers 1 patient and returns the central db patient id

## C++ Classes

### TBICDSClientView.cpp

Code Type: C++ Class

Uses PL/SQL Procedures: N/A

Uses MDWS Calls: N/A

Lines of Code: 121

* void CTBICDSClientView::CPRSInit()
  + US:5712 called when the app is activated from the CPRS tools menu

## Oracle PL/SQL

### pck\_fx\_sec.bdy

Uses Tables: FX\_SEC, FX\_SESSION, FX\_SESSION\_VALUE, FX\_USER, FX\_USER\_ID, FX\_RESTRICTED\_NAME, UTL\_DECSTR, FX\_IPLOGIN\_ATTEMPTS, FX\_PAGE\_ACCESS, FX\_USER\_RIGHTS, FX\_AUDIT, UTL\_SEC.ENCRYPTDATA, FX\_SEC.UPDATESUATUSERRIGHTS, FX\_USER\_ID), STAT\_SEC\_QUESTIONS, FX\_SEC.GETSECURITYQUESTIONS, FX\_SEC\_QUESTIONS, FX\_SEC.GETUSERQUESTIONS, FX\_SEC.UPDATESECQUESTIONS, FX\_SEC.CHECKSECURITYQUESTIONS

Lines of Code: 1993

* procedure CreateSession(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vWebSessionID in varchar2,

po\_vDBSessionID out varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + creates a database session for a user. used after a successful login to an external system such as MDWS to establish a session for the user in the database
* procedure ValidatePassword(pi\_vKey in varchar2,

pi\_nUserID in number,

pi\_vUserName in varchar2,

pi\_vOldPassword in varchar2,

pi\_vPassword in varchar2,

pi\_vCOldPassword in varchar2,

pi\_vCPassword in varchar2,

pi\_vCUserName in varchar2,

pi\_nResetPassword in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + validate the password against rules
* procedure DeleteAllSessionValues(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + delete all session values
* procedure DeleteSessionValue(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + delete 1 session value
* procedure GetSessionValue(pi\_vDBSessionID in varchar2,

pi\_vWebSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

po\_vKeyValue out varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + get a session value
* procedure SetSessionValue(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

pi\_vKeyValue in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + set a session value
* procedure CertLogin(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number, not used but passed in to maintain consistant default param list

pi\_vCert in varchar2,

po\_nUserID out number,

po\_vDBSessionID out varchar2,

po\_nTimeout out number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + cac cert login
* procedure ChangePassword(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number, not used but passed in to maintain consistant default param list

pi\_vKey in varchar2,

pi\_vUserName in varchar2,

pi\_vOldPassword in varchar2,

pi\_vPassword in varchar2,

pi\_vCert in varchar2,

pi\_vCOldPassword in varchar2,

pi\_vCPassword in varchar2,

pi\_vCUserName in varchar2,

po\_nUserID out number,

po\_vDBSessionID out varchar2,

po\_nTimeout out number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + change password, this will also logg the user in
* procedure Sign(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vUserName in varchar2,

pi\_vPassword in varchar2,

po\_vProviderID out varchar2,

po\_nUserType out number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + used for signing notes etc.
* procedure Login(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number, not used but passed in to maintain consistant default param list

pi\_vUserName in varchar2,

pi\_vPassword in varchar2,

pi\_vCert in varchar2,

po\_nUserID out number,

po\_vDBSessionID out varchar2,

po\_nTimeout out number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + login this is the version that requires a username and password after a successful login the persons CAC will be associated with this account
* procedure AuditPageAccess(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vPageName in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + audit page acceess
* procedure AuditTransaction(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vSPName in varchar2,

pi\_clAuditXML in clob,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + audits a transaction by encrypting the clob passed in and storing the params and clob in the fx\_audit table
* procedure LogOff(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + Log Off, clears session from the db
* procedure GetFXUserRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vEncUID in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* + gets an fx\_user record given the encrypted uid
* procedure InsertFXUser(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

pi\_vProviderID in varchar2,

pi\_vUserName in varchar2,

pi\_vPassword in varchar2,

pi\_nAccountLocked in number,

pi\_nAccountInactive in number,

pi\_vCOldPassword in varchar2,

pi\_vCPassword in varchar2,

pi\_vCUserName in varchar2,

po\_nFXUserID out number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + insert a record into the fx\_user table and update the fx\_user\_id in the suat\_user table
* procedure UpdateFXUser(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nFXUserID in number,

pi\_vProviderID in varchar2,

pi\_vUserName in varchar2,

pi\_vPassword in varchar2,

pi\_nAccountLocked in number,

pi\_nAccountInactive in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + update an fx\_user record
* procedure UpdateFXUserRights(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nFxUserID in number,

pi\_nUserType in number,

pi\_nUserRights in number,

pi\_nUserReadOnly in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + update fx\_user\_rights
* procedure UpdateFXUserPWD(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

pi\_nFXUserID in number,

pi\_vUserName in varchar2,

pi\_vPassword in varchar2,

pi\_nAccountLocked in number,

pi\_nAccountInactive in number,

pi\_vCPassword in varchar2,

pi\_vCUserName in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + update an fx\_user record pwd
* procedure UpdateFXUserOptions(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nFXUserID in number,

pi\_nAccountLocked in number,

pi\_nAccountInactive in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + update an fx\_user options
* procedure GetFXUsernamePasswordRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vProviderID in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* + gets an Username and Password given the providerid
* procedure GetFXUserIdRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vProviderID in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* + gets an FXUserID given the providerid
* procedure CheckFXUserRecRS(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vProviderID in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* + gets a FXUserID given the providerid
* procedure GetSecurityQuestions(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nQuestionGrp in number,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* + gets security questions
* procedure GetUserQuestions(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vUsername in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2,

rs out RetRefCursor)

* + gets user questions
* procedure UpdateSecQuestions(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nQuestionID\_1 in number,

pi\_vAnswer\_1 in varchar2,

pi\_nQuestionID\_2 in number,

pi\_vAnswer\_2 in varchar2,

pi\_nQuestionID\_3 in number,

pi\_vAnswer\_3 in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + update security questions
* procedure CheckSecurityQuestions(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_nQuestionID\_1 in number,

pi\_vAnswer\_1 in varchar2,

pi\_nQuestionID\_2 in number,

pi\_vAnswer\_2 in varchar2,

pi\_nQuestionID\_3 in number,

pi\_vAnswer\_3 in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + check security questions
* procedure ResetPassword(pi\_vSessionID in varchar2,

pi\_vSessionClientIP in varchar2,

pi\_nUserID in number,

pi\_vKey in varchar2,

pi\_nFXUserID in number,

pi\_vUserName in varchar2,

pi\_vPassword in varchar2,

pi\_vCPassword in varchar2,

pi\_vCUserName in varchar2,

po\_nStatusCode out number,

po\_vStatusComment out varchar2)

* + reset password