# Introduction

I have supplied you with this readme.doc that explains the xml file you will be working with and some areas to focus on for the first phase of the project.

# The Form

The xml file (BH\_Progress\_Assessment.xml) will generate a form like Figure 1. It is a version of a behavioral health assessment form we use.

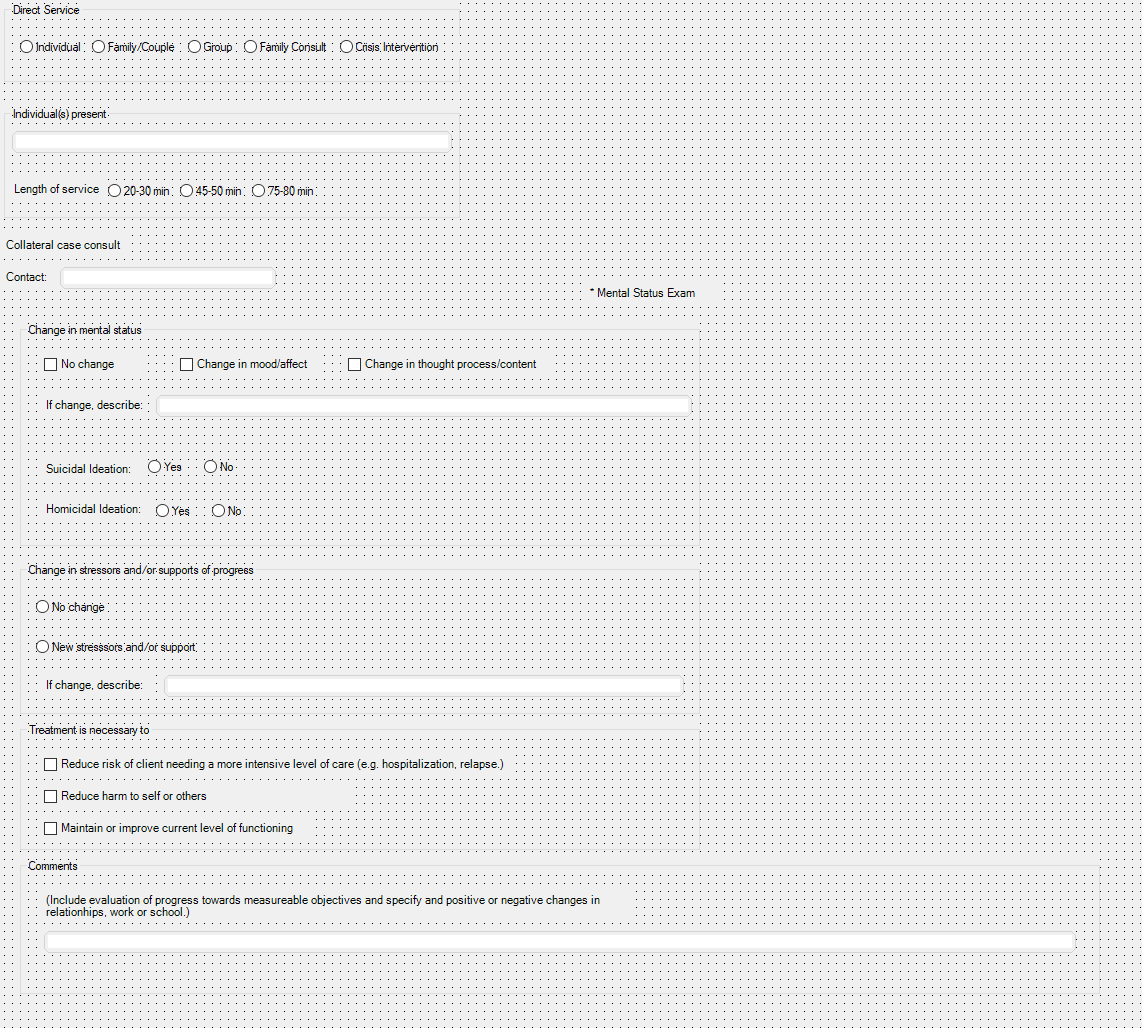


Figure 1

# The XML File

I suggest you use NotePad++ (or your favorite xml editor) to open the xml file, BH\_Progress\_Assessment.xml so you can read it. The file contents should look like Figure 2 when it is opened.

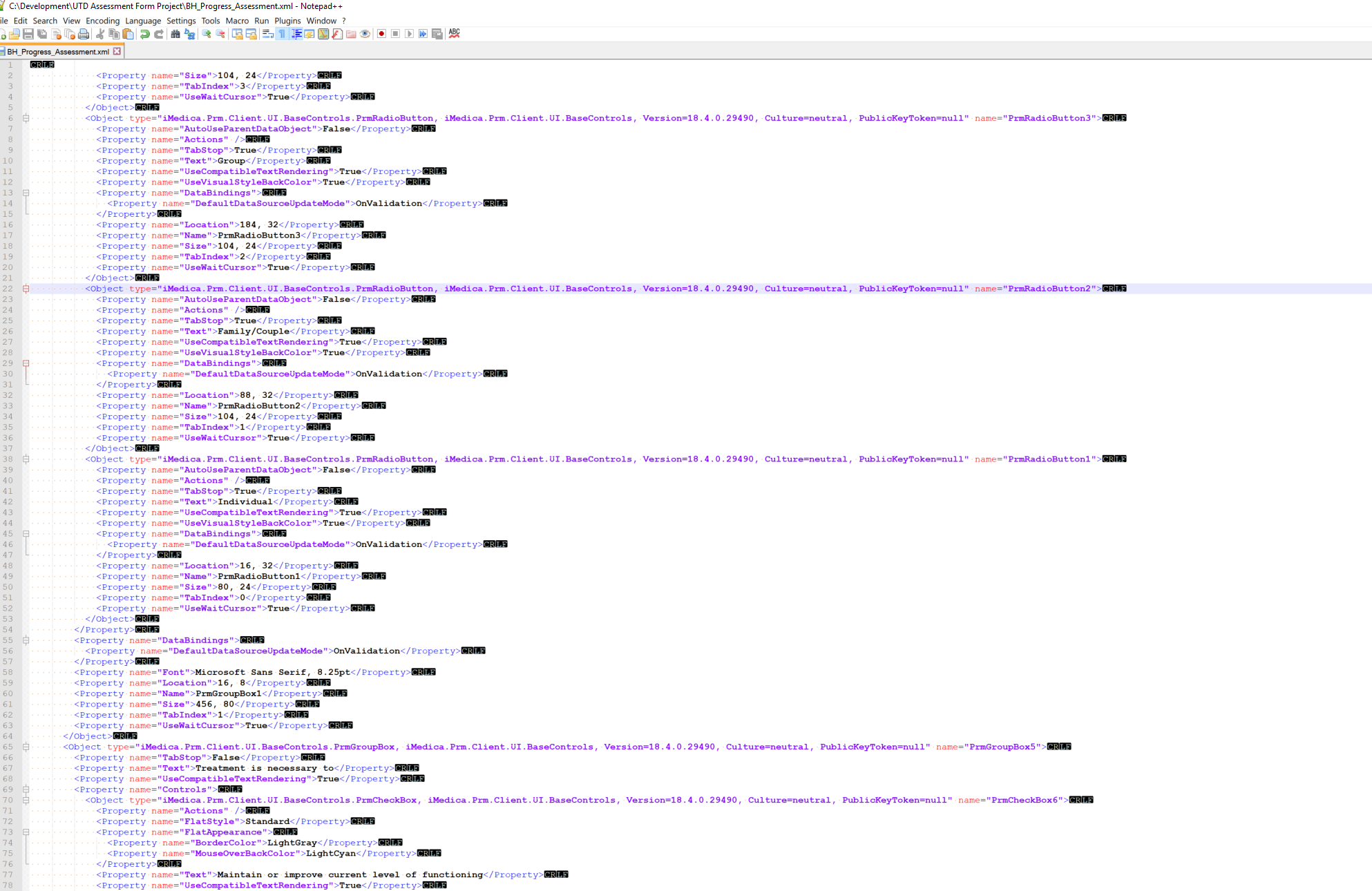


Figure 2

An example UI element (also known as a control) is defined in a block such as:

*<Object type="iMedica.Prm.Client.UI.BaseControls.PrmRadioButton, iMedica.Prm.Client.UI.BaseControls, Version=18.4.0.29490, Culture=neutral, PublicKeyToken=null" name="PrmRadioButton3">*

*<Property name="AutoUseParentDataObject">False</Property>*

*<Property name="Actions" />*

*<Property name="TabStop">True</Property>*

***<Property name="Text">Group</Property>***

*<Property name="UseCompatibleTextRendering">True</Property>*

*<Property name="UseVisualStyleBackColor">True</Property>*

*<Property name="DataBindings">*

*<Property name="DefaultDataSourceUpdateMode">OnValidation</Property>*

*</Property>*

*<Property name="Location">184, 32</Property>*

***<Property name="Name">PrmRadioButton3</Property>***

*<Property name="Size">104, 24</Property>*

*<Property name="TabIndex">2</Property>*

*<Property name="UseWaitCursor">True</Property>*

*</Object>*

It refers to a radio button that has the name of *PrmRadioButton3.* It is the third radio button in the Figure 1 with a label of “Group”. Each control on the form is defined in this same manner.

# Initial Goals

As we discussed, in our kickoff, an initial goal is to be able to accomplish the following:

1. Be able to open the xml file
2. Parse the xml to identify the controls
3. Dynamically generate html that contains the control as its equivalent html element type

As you progress, you will be able to then move on to more advanced topics such as putting the xml into a database than accessing the data through a web service via a REST endpoint. But to get to a jumping off point, start out in small steps. You can use node.js to facilitate the operations on the xml file such as opening it & identifying the controls. You will eventually be able to build up from there to creat the html.

When you get to the html generation portion, you will have to study up on HTML and how to dynamically create elements such as input boxes, radio buttons etc… Then you will gradually get into displaying the HTML in your angular 7 front end client.

As you progress later on with needing to create the backend web service, you will get into the MySql issues for database access. But those issues will be addressed later on in the project.