Function List:

* User Sees list of exercises
  + Enumerate list of exercises
    - Get exercise name
    - Get exercise good form skeleton for thumbnail(?)
  + Create/Modify/Delete Exercises routines
    - Create/Modify/Delete classifiers routines for each exercise
* Select an exercise
  + Load specific exercise data (including full training data)
  + Train machine learning algorithm based on data
* Start doing reps
  + Detect when a user has paused (I.e: extended or contracted in their rep)
  + Get the form critique for the current identified position
    - **What is this in the form of?**
    - **What should be displayed?**
    - **Should it be text?**
    - **Should it be an arrow pointing to an accepted position?**
  + Save skeletons of each rep to a list to enumerate at the end of use
* Finish their set
* Sees detailed advice on form improvements
  + Enumerate a list of skeletons of all reps done in that set
    - **How to store this data? Should be as a Body or bodyDouble? (most likely body as memory isn’t too much of a concern)**
  + Based on User Input save a skeleton as training data with classifier
    - Include create/modify/delete classifiers here

Rough API Map:

SpotMeML.cs

* Initialize machine learning based on exercise
* Detect when a user has paused
* Get form critique for an inputted skeleton

Exercise.cs

* Get/Set exercise name
* Get/Set exercise good form (contracted and extended)
* Get/Set list of classifiers

Classifier.cs

* Get/set classifier name

ExerciseFileIO.cs (Could be static?)

* Save Exercise
* Load Exercise
* Delete Exercise
* Modify Exercise (Could be done by just giving access to exercise and then saving it)
* Add classifier
* Modify classifier
* Delete classifier
* Enumerate Exercise List (Include names, and good form skeleton for thumbnail)

MainWindow.cs (The form for displaying the window)

* Display skeleton in small thumbnail fomats

Feedback.cs

* Save skeleton of that rep
* Get/Set list of rep skeletons done in that set
* POSSIBLE EXTENSION: Only get list of skeletons done with bad form, etc.