We have chosen RiotWatcher, a wrapper for the Riot Game API for League Of Legends, as our new project.

Our 25 test cases are derived from the following 5 methods:

static_get_item
static_get_summoner_spell
static_get_mastery
static_get_champion
static_get_rune

Currently, our testing framework is fully functional and complete. We have designed our twenty five test cases around the above methods. We used these because we thought they were easiest, and we were familiar with some of the subject material of what the methods deal with (what a champion is, what an item is, etc.) so that we would be able to easily determine if we were getting funky or odd results. Along with this, we developed a few extra test cases to ensure that the information was correct, and in the event that some of the data changes in the coming days.

While we do not believe the data will change and don't think our driver will be affected at all by it, a new season of League of Legends has just begun, and therefore Riot is in the process of evaluating items, runes, and masteries and evaluating whether or not they should stay in the game. This information has been made pretty open to what they intend on changing, and therefore we're not very worried about it, but we developed the extra test cases just in case.

Even though our framework is complete in earnest and we have our test cases developed, there are still a few minor things that our group intends to address before we complete the final version of it. One of the first things we need to finish up on is documenting our code. It is crucial that we have documentation explaining how everything works, even though we went to great strands to attempt naming and writing our code in a way to make it best and most easily readable.

Next, we need to create better formatting for our HTML report of the results. Currently, the headers are repeated on each row instead of just once for the entire table. This is obviously not what we intend to happen, and is a very simple fix (a problem in logic of one of our for loops, most likely). It should be an easy fix and something that we will easily be able to implement.

Finally, we need to work on getting the folder structure to work. Again, this is a simple change as we just need to work with the path of the system. But, so far we just had everything running from the same folder. We already have a plan in place to complete this fix.

Beyond that, our group is still working great! However, we're all getting bogged down from the end-of-the-year rush of classes, finals, and projects, so it will be interesting to see how we push forward to complete the final stint of our project!