**Main project Objectives**

* Stable working website
* Play a fantasy game
* Social media aspects to it
* User can register
* Ads can be placed on the site
* Instant messaging service
* Forum service
* Scores update using RSS feed
* Player score updated automatically on login
* Admin can log in to do maintenance

**Success Criteria**

* User should be able to register
* User should be able to select a team of 8 players
* User should be able to user instant messaging service to chat
* User can post on a forum
* Users points automatically updated each week
* User can chat with others near using a GPS system

**Management Deliverables:**

Feasibility Study and Project plan: A feasibility study will be conducted so that a decision on the viability of the project can be made. This should contain an outline of the project along with the benefits, requirements and alternatives if any.

Analysis and design document: This document will contain all the requirements and functions that the Application should have. This will contain our UML diagrams which will plot how the user interacts with the software and what extra functions algorithms we might need.

Source code: We will use the Analysis and Design document to build the code for our software. This file should contain all code, images and installation files.

Testing Document: All tests performed on the software will contained in this. Testing will done in all stages of the application and be documented for further record.

**Technical Deliverables:**

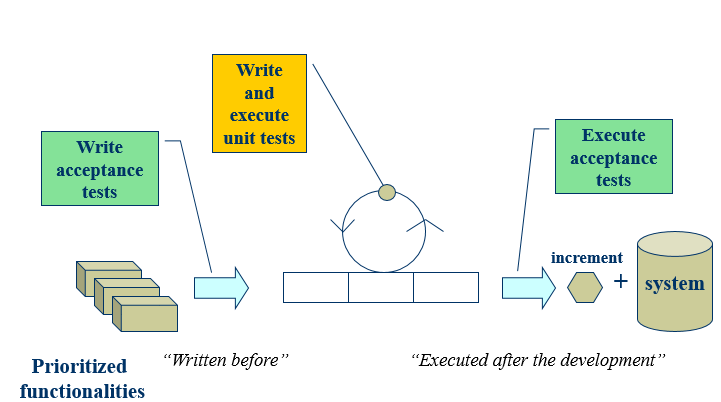
* A clear and easy way for a user to navigate through and complete the team selection
* User login to access an account or set up an account on first use.
* User should be able to change a player once a week
* Administrator login to change variables without interfering with ongoing game
* Database to store account details.
* Real time update of player scoring using RSS feeds
* Instant message style chat built in
* Forum messages used for discussion of games
* User able to join leagues and check overall standing

## **Testing**

Testing is the most important part of any software project. Eradicating errors and bugs is vital to the user experience. We came up with some main components to our testing phase.

* Acceptance Testing – this checks if the overall system is functioning as required.
* Unit testing – this is basically testing of a single function, procedure, class.
* Integration testing – this checks that units tested in isolation work properly when put together.
* System testing – here the emphasis is to ensure that the whole system can cope with real data, monitor system performance, test the system’s error handling and recovery routines.
* Regression Testing – this checks that the system preserves its functionality after maintenance and/or evolution tasks.

This testing phase is highly iterative and can be very effective in creating a bug free user experience.

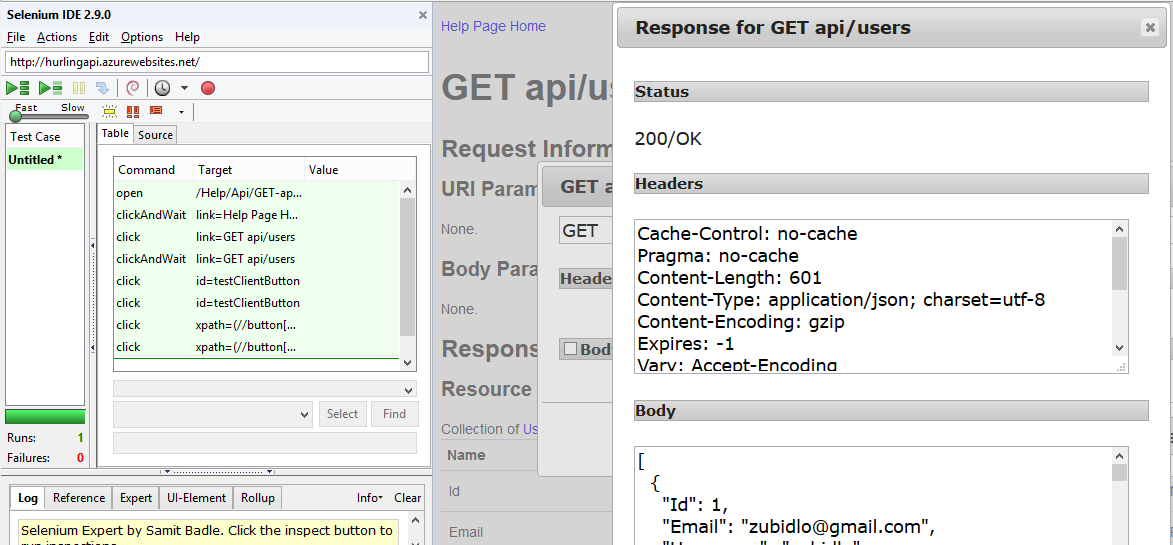
****

Testing the API

We tested the API using Selenium IDE for Firefox. Selenium is an automated web testing tool that allows the user to navigate and interact with a website, and have Selenium track those navigations and changes. You can then choose to let Selenium re-run these while looking for any inconsistencies or errors. The first step was to install the Selenium IDE for Firefox. We then began to test the API help page. This page essentially interacts with the API in the backend and simulates to query and retrieval of the JSON data. Any errors would show up in selenium.

**Test 1**

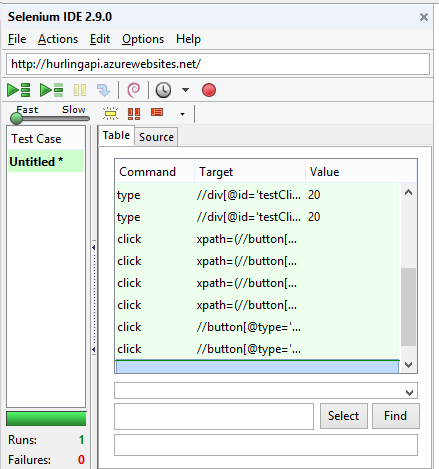
The first test was to try and retrieve all users from the API. This was done by simply setting the URL in Selenium to the main help page URL. Then we navigated to the users and performed the query as normal. The query was then re-run in Selenium.



We can see that the test passed successfully. This means the navigation and retrieval of the JSON data from the API was error free.

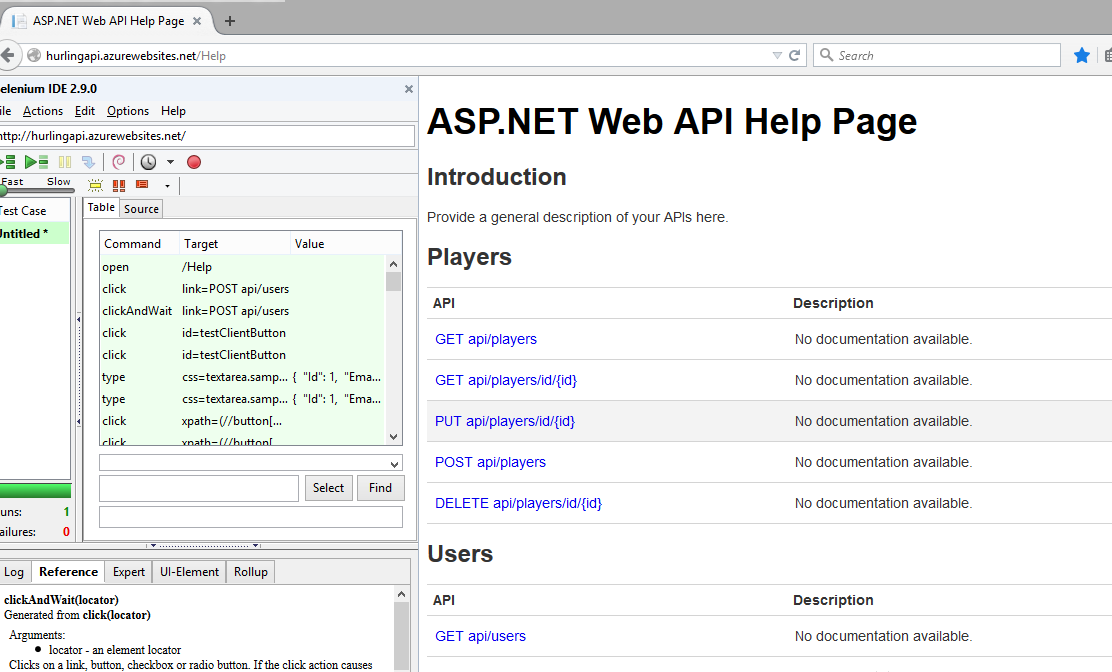
**Test 2**

The second test was to input a player into a specified user’s team via the API. We simulated the deleting of player with ID 20 from the team with ID 2. We then deleted the player so that when Selenium ran, it would not get an error trying to re-add a player that is already in a team! As we can see the test was successful and we received no errors.



**Test 3**

The final test we ran on the API was a little more complex. It involved adding a new user, creating a team for him, then adding a player to that team. We created a new user, then a new team, and added player with ID 1 to that team, who’s ID was 1061. The user ID was 1030. The test ran successfully and we encountered no errors.



**Testing the application Website**

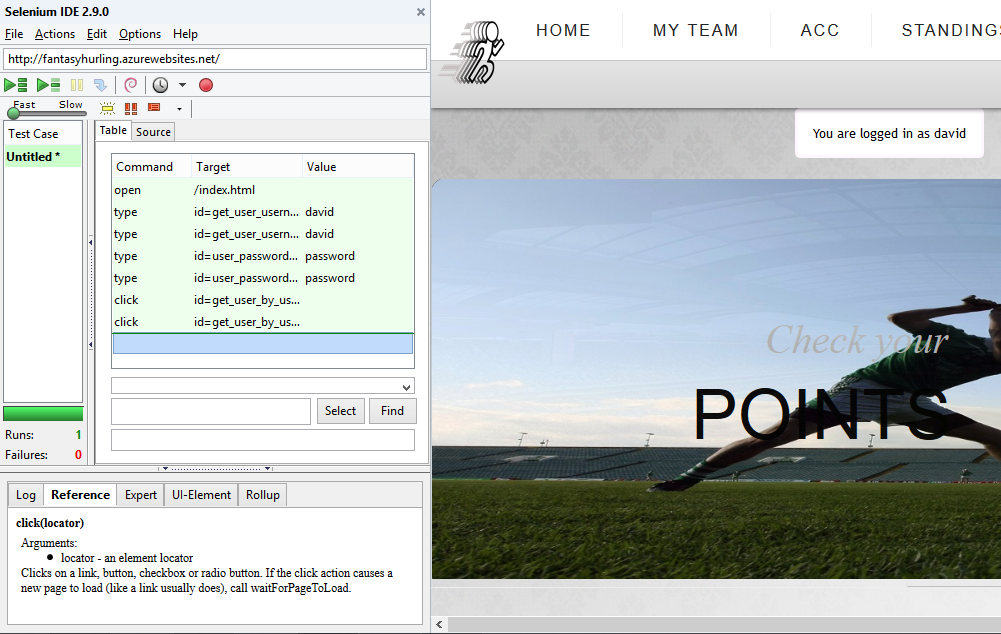
In regards the testing of the site of the site itself, there were a few factors deemed as very important and were considered to be so vital as even a small failure was unacceptable. These were as follows:

* The login of the user should be flawless
* Registration should be flawless
* Changing user details
* Deleting a player
* Adding a player

Due to the high level of importance for all of these interactions with the site, we decided to perform Selenium tests on all 5 scenarios.

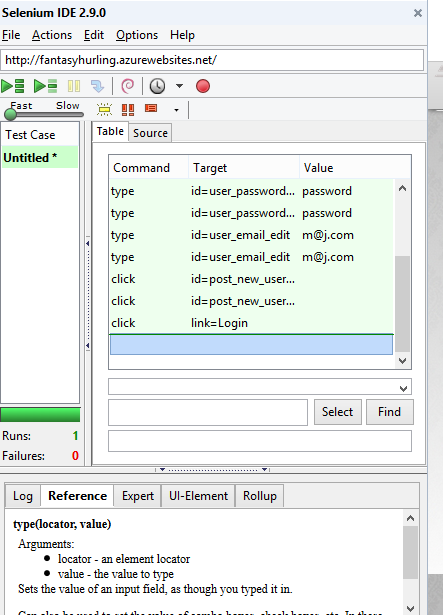
**Test 1**

The first test was to see if the user could login properly and without an errors. We ran the Selenium test and we can see from the results below the test was successful with no errors:



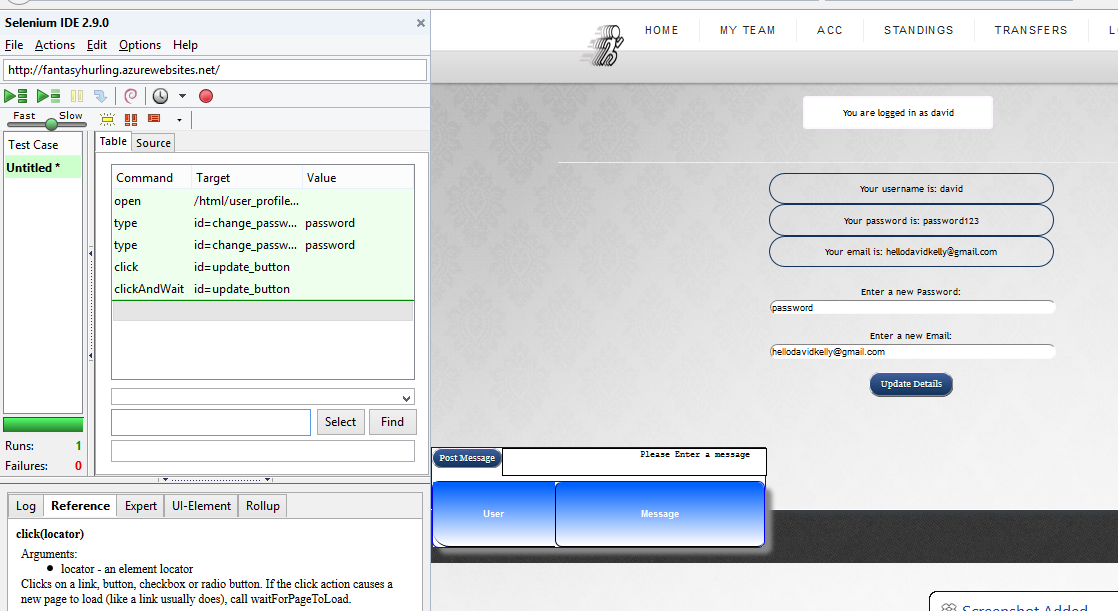
**Test 2**

The second test we ran was to make sure the user could register with no issues. Again we ran the selenium test and as expected there were no problems that needed to be resolved. We can see the output of the successful test below:



**Test 3**

The next test was to see if the user could change their details without issue. We ran the Selenium test and the results were again as expected, the test was successful with no errors. We can see the output below.



**Test 4**

We next ran a test to see if the user could delete a player from his team. The results were expected to be successful with no failures and this was indeed the case.



**Test 5**

The last test was to see if we could add a player into the team, the results were as expected and with no failures. We removed player with position 2 and the re-added him capturing the input with Selenium. We then stopped the capture, removed him and used Selenium to try automate his addition to the team again. It worked with no issues.



## **Conclusion**

We did not choose a trivial project. Being unexperienced developers we didn’t foresee the complexity and challenges involved in developing a fantasy sport web application. After realizing the scope of the project we changed our initial ideas how to approach this project development. We decided to follow prototyping methodology implementing and test one functionality at the time. Hoping to learn from each iteration and deliver a modern optimized rich user experience application.

In this occasion we decided to use the most modern technologies and designs. We are building a Web API Service so that different clients could get access to CRUD methods. This way the additional clients can get developed easily. For example in the future we can easily implement Android client or Windows 8 metro client for our fantasy hurling application. We are using Entity Framework ***[14]*** to automatically map database table rows to entity objects so we don’t need to write one sql query. We are implementing our Web API using .NET Web API 2 framework which is powerful toolkit capable of asynchronous responses, cross-origin requests and much more. We are deploying the application up on Microsoft Azure Cloud.

This project is an opportunity for us to learn new technologies and get valuable firsthand experience and skills in web development.