AccuLynx Coding Challenge - Overview

This document is a quick overview of what was done with the coding challenge that was provided.

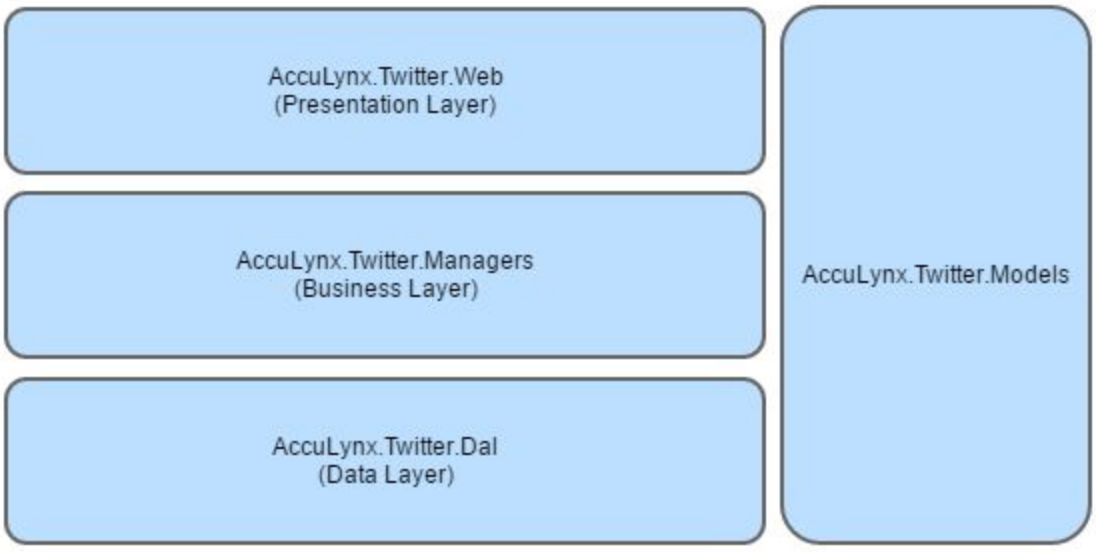
Primary Technologies:

- Entity Framework
- Bootstrap
- AngularJS
- MVC
- SQL Server

Key Resources:

- http://www.entityframeworktutorial.net/
- https://docs.microsoft.com/en-us/azure/app-service/app-service-web-get-started-dotnet-framework
- http://curran.github.io/screencasts/introToAngular/exampleViewer/#/
- https://www.codeproject.com/Articles/1200390/Taming-the-Twitter-API-in-Csharp

Below is a diagram that explains the different projects in the solution.



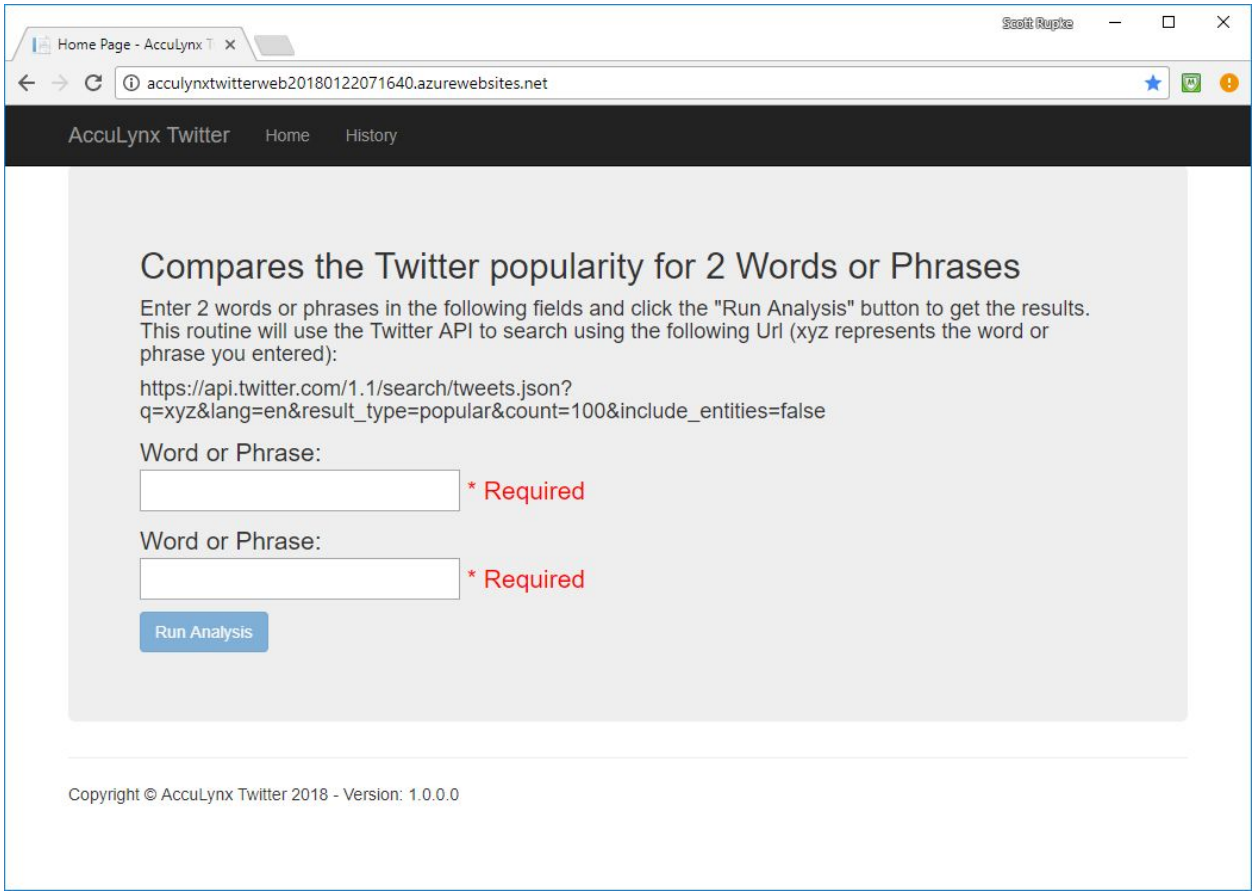
Below is the location of the website published to Azure.

http://acculynxtwitterweb20180122071640.azurewebsites.net/

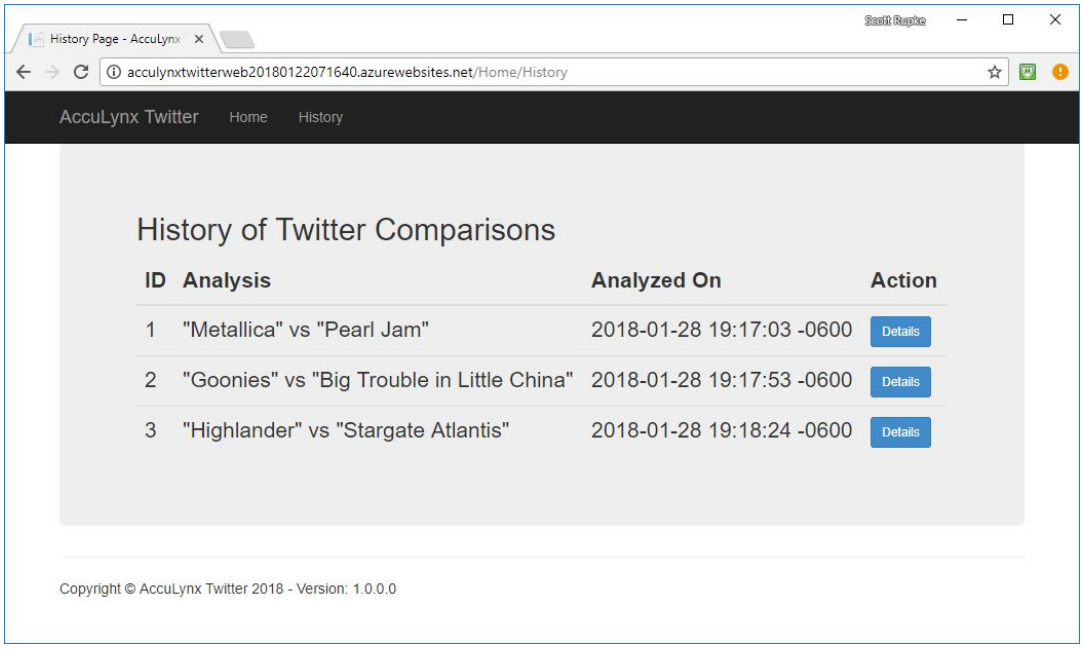
Below is the location of the Visual Studio solution in GitHub.

https://github.com/srupke/AccuLynx

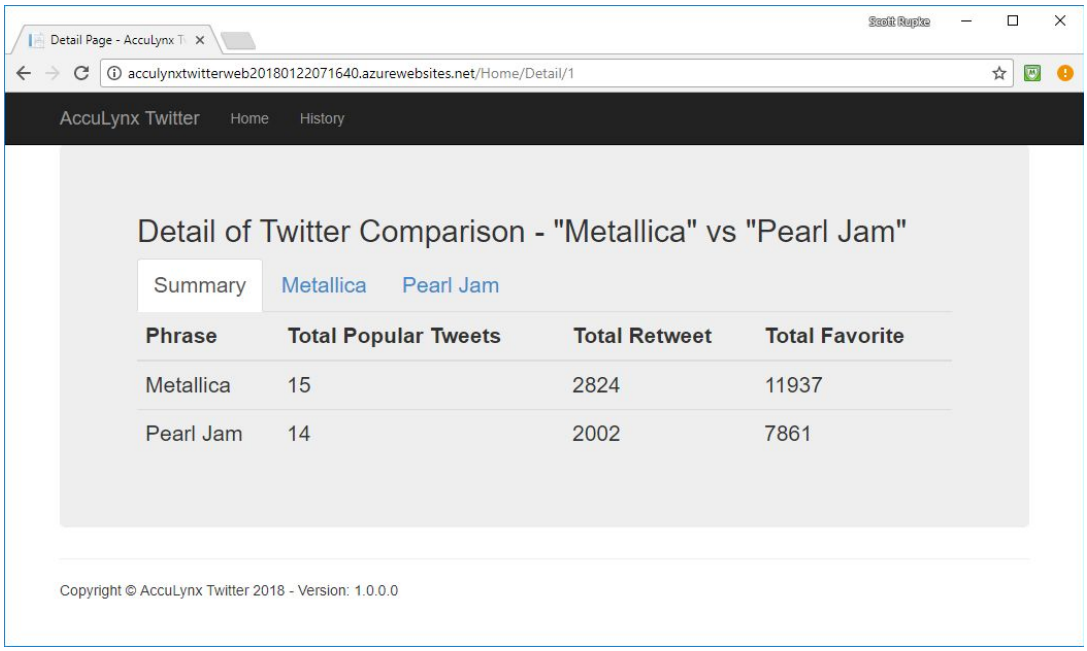
Below is a screenshot of the Home page of the web application. This page explains what to do to start an analysis of two words or phrases. In the header you can click on the History option to see previously run comparisons.



Below is a screenshot of the History Page of the web application. This page is a list of previously run comparisons.



Below is a screenshot of the Detail Page of the web application. Notice that there are three tabs in the display. The first tab is a summary of information from the next two tabs showing which word or phrase is more popular.



The following is the script used to create the database. I used Entity Framework and the Code-First concepts.

```
CREATE TABLE [dbo].[TwitterAnalysis] (
    [AnalysisId] [int] NOT NULL IDENTITY,
    [Description] [nvarchar](max) NOT NULL,
    [AnalyzedOn] [datetime] NOT NULL,
    [RowVersion] rowversion NOT NULL,
    CONSTRAINT [PK_dbo.TwitterAnalysis] PRIMARY KEY ([AnalysisId])
)
CREATE TABLE [dbo].[TwitterAnalysisPhrase] (
    [PhraseId] [int] NOT NULL IDENTITY,
    [OrderId] [int] NOT NULL,
    [PhraseText] [nvarchar](max) NOT NULL,
    [AnalysisId] [int] NOT NULL,
    [RowVersion] rowversion NOT NULL,
    CONSTRAINT [PK_dbo.TwitterAnalysisPhrase] PRIMARY KEY ([PhraseId])
)
CREATE INDEX [IX_AnalysisId] ON [dbo].[TwitterAnalysisPhrase]([AnalysisId])
CREATE TABLE [dbo].[TwitterAnalysisPhraseDetail] (
    [AnalysisPhraseDetailId] [int] NOT NULL IDENTITY,
    [User] [nvarchar](max) NOT NULL,
    [ScreenName] [nvarchar](max) NOT NULL,
    [TweetCreatedAt] [datetime] NOT NULL,
    [TweetText] [nvarchar](max) NOT NULL,
    [TweetTextTruncated] [bit] NOT NULL,
    [RetweetCount] [int] NOT NULL,
    [FavoriteCount] [int] NOT NULL,
    [PhraseId] [int] NOT NULL,
    [RowVersion] rowversion NOT NULL,
    CONSTRAINT [PK_dbo.TwitterAnalysisPhraseDetail] PRIMARY KEY ([AnalysisPhraseDetailId])
)
CREATE INDEX [IX_PhraseId] ON [dbo].[TwitterAnalysisPhraseDetail]([PhraseId])
ALTER TABLE [dbo].[TwitterAnalysisPhrase] ADD CONSTRAINT
[FK_dbo.TwitterAnalysisPhrase_dbo.TwitterAnalysis_AnalysisId] FOREIGN KEY ([AnalysisId]) REFERENCES
[dbo].[TwitterAnalysis] ([AnalysisId]) ON DELETE CASCADE
ALTER TABLE [dbo].[TwitterAnalysisPhraseDetail] ADD CONSTRAINT
[FK_dbo.TwitterAnalysisPhraseDetail_dbo.TwitterAnalysisPhrase_PhraseId] FOREIGN KEY ([PhraseId])
REFERENCES [dbo].[TwitterAnalysisPhrase] ([PhraseId]) ON DELETE CASCADE
CREATE TABLE [dbo].[__MigrationHistory] (
    [MigrationId] [nvarchar](150) NOT NULL,
    [ContextKey] [nvarchar](300) NOT NULL,
    [Model] [varbinary](max) NOT NULL,
    [ProductVersion] [nvarchar](32) NOT NULL,
    CONSTRAINT [PK_dbo.__MigrationHistory] PRIMARY KEY ([MigrationId], [ContextKey])
)
```