

# May 13-31

## News Streams

Members: Jamison Bradley, Lance Staley, Charles Litfin

Advisor: Srikanta Tirthapura

Client: IBM Rochester

# Problem

- Several different news sources all publish articles about the same topics.
- Causes information overload for users.

# Solution

- Aggregation website for news articles.
- Match articles about same story together for user.
- Use IBM Infosphere Streams on backend to help with aggregation.

# System Architecture

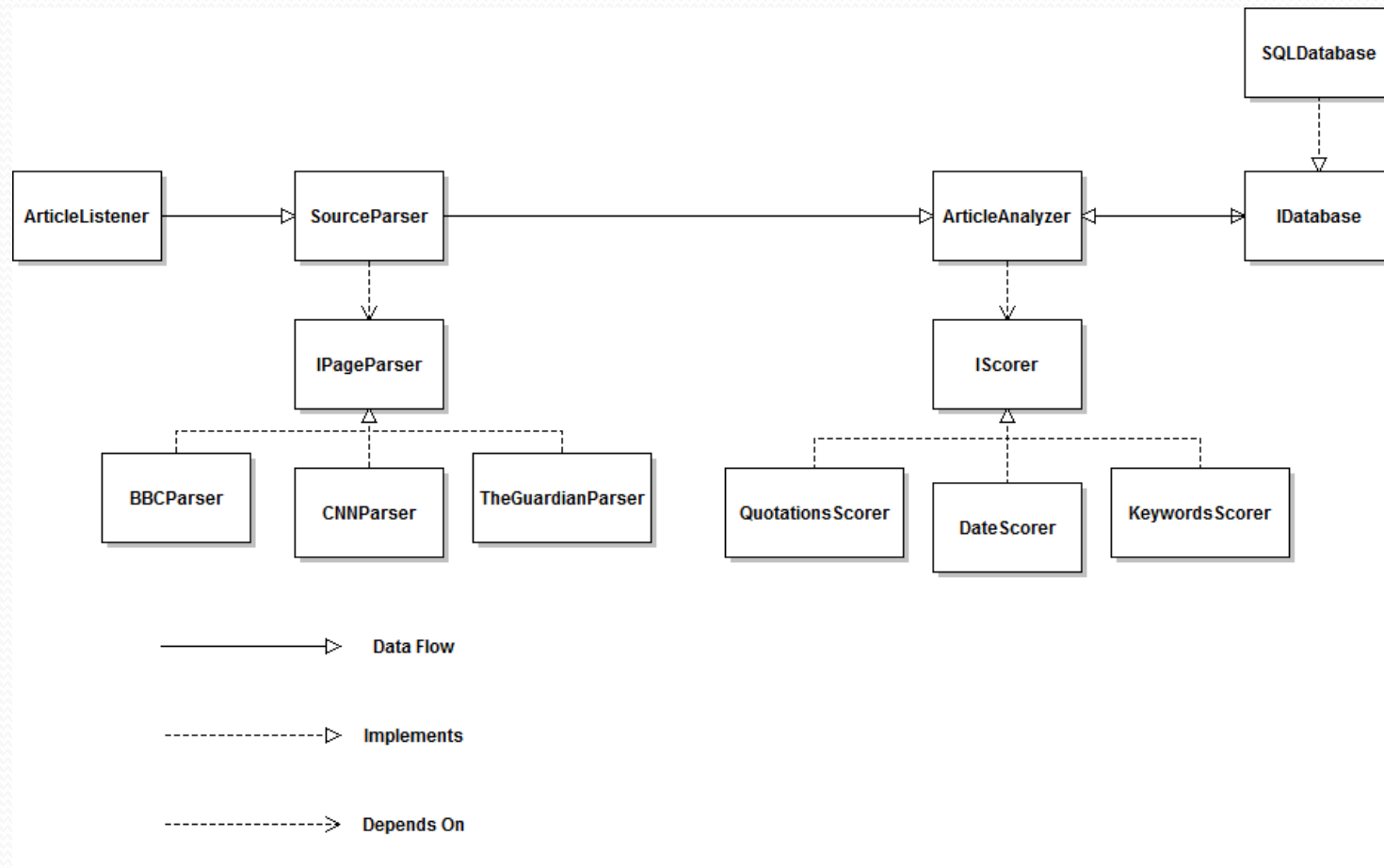
## Dataflow Diagram



## Functionality

- User Frontend – A website that allows the user to easily browse news stories.
- Datastore – We will use a MySQL Database for storing the state of the system.
- Analysis and Aggregation – IBM Infosphere Streams and Java.

# Backend Design



# Database Tables - Article

Field	Type	Description
Article ID	Int	The id that is associated with the article, a unique id will be given to every article. This will be the primary key of the Article Table.
StoryID	Int	The id that is associated with the story the article is talking about, if several articles are about the same story they will have the same id. This will be the primary key in the Story table.
URL	String	The url of the article.
Source	String	The name of the news source that the article is from.
Date	Long	The date of publication for the article.
Title	String	The title that was given to the article by the news source.
Article Keywords	String[]	The key words that were found in the article.
Category	String	The category that the news source placed this article in on its website.
Quotations	String[]	Quotations that were found in the article.

# Database Tables - Story

Field	Type	Description
Story ID	Int	See description in Article Table.
Category	String	The category that the story falls under on our website.
Date	Long	The date that the particular story occurred on, it will be an average of all the articles that match to this story, and will be the date we use on our website.
Title	String	Will be the title of the story that we display on our website.

# Database Tables - Users

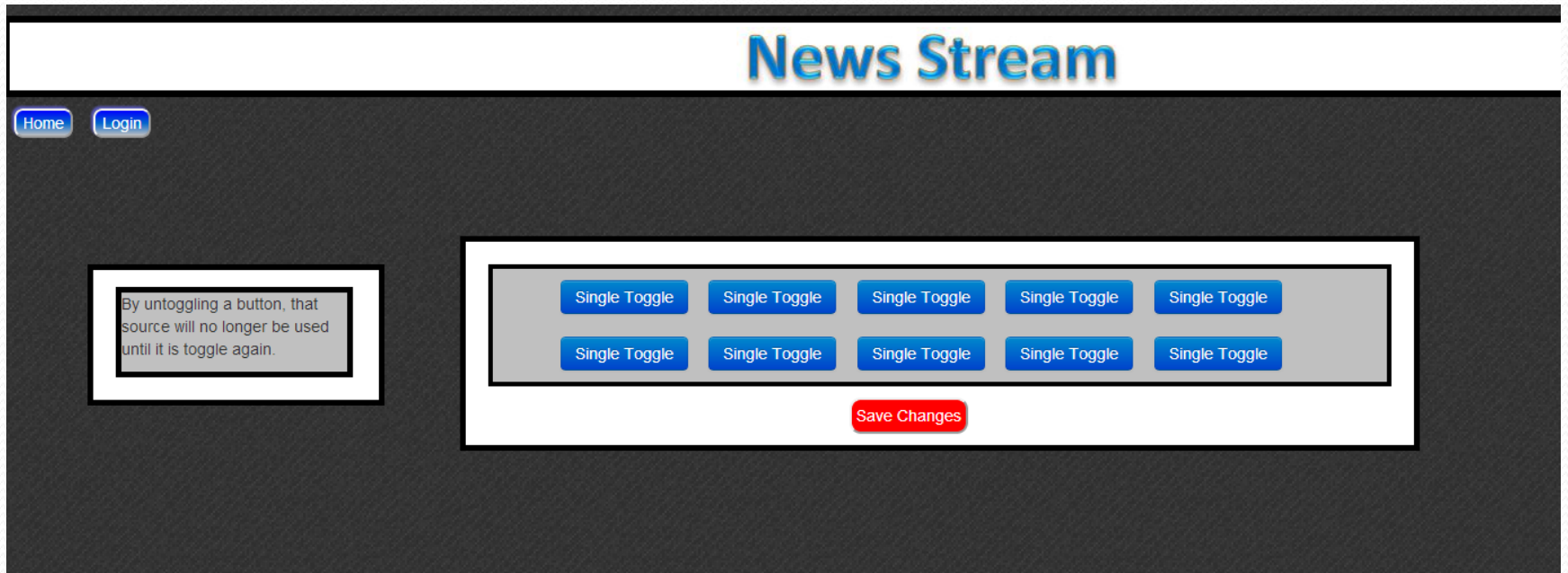
Field	Type	Description
User Name	String	Username of the person using our service.
Source Exceptions	String[]	Sources they do not wish to view.
Password	String	Hashed version of user's password.



# Frontend Progress



# Frontend Progress



# Frontend Progress

## News Stream

[home](#)

[Customize Sources](#)

User name:

Password:

[Login](#)

# Technical Challenges

- Facebook Login – Storing user preferences based on login username appears to violate Facebook’s terms of use, so we aren’t going to risk using it.
- Data types- Types of data that can be stored in the database are limited, and thus must be worked around.
- Streams – Didn’t get requested examples till a couple months after request was made.

# Test Plan

- Will vary from module to module
  - Unit Testing – For critical pieces of the backend modules and database interaction.
  - Black Box Testing – When doing unit testing some of our tests will rely on our knowledge of how the code is working to test it.
  - Usability Testing – We'll test the front-end functionality by trying to replicate how a potential user of the system would use the service and making sure correct information is shown when desired by the user.

# Questions?