Web-based Newsgroup (NewW-B)

Description:

We will implement a web based newsgroup platform that will users to browse newsgroup articles. The platform will include a user management and authentication system, a topic based subscription service, a ranking and comment system, a browsing interface that prioritizes highly rated articles, and both simple and advanced search tools.

Features

- Download relevant articles
  - Use RSS feeds to downloads html articles
  - Sort articles into the file system
  - Input article records into relevant DB tables
- Manage User Account
  - Create and account user profiles
  - Mange usernames and passwords
  - Verify login details
- Subscription service
  - Subscribe and unsubscribe from news categories
- Browsing Service
  - Decide what articles and in what order to display to the user
  - Update users view articles history
  - Allow filtering by category, date, etc
- Search
  - Simple search
  - Advanced search with filtering

Implementation

We will construct a NoSQL JSON Document Database with the following document model:

```
const UserSchema = new Schema({
    u_id : {type: String, required: true},
    f_name : String,
    l_name : String,
    email : String,
    pw : {type: String, required: true},
    create_date : String,
    rank : Number,
    follows : [Tag.schema],
    favorites : [mongoose.Schema.Types.ObjectId],
    voted_on : [
        {article: mongoose.Schema.Types.ObjectId,
         vote : Number}],
    comments : [mongoose.Schema.Types.ObjectId]
});

const ArticleSchema = new Schema({
    title : String,
    author : String,
    date : String,
    URL : String,
    tags : [Tag.schema],
    rank : Number,
    comments : [Comment.schema]
});

const CommentSchema = new Schema({
    u_id : {
        type: mongoose.Schema.Types.ObjectId,
        required: true
    },
    date : String,
    txt : String,
    rank : Number
});

const TagSchema = new Schema({
    tag:String
});
```

We selected NoSQL as our DB language because we need the ability to scale the number of users and the number of articles available to users. This design also allows inserting new articles and users without the need to populate all attributes and perhaps to add new attributes as we expand the functionality of the newsgroup system. We will use MongoDB as our DBMS running on a server located in our lab.

The front end of our platform will be a web-app created in JavaScript and using the Node.js framework for dynamically querying the DB and updating pages.

Testing

1. Mocha unit test
    a. Method tests:
        i. create_user_test(), login_test(), subscribe_test(), vote_test(), search_test, etc
    b. Multi-user test:
        i. Simulate actions of a group of random users interacting with the system(i.e. subscribing to categories, reading articles, searching, voting, and commenting) while timing the various user interactions.