Autism Searches: A Modern Search Engine for ASD Related Topics

Joshua Schappel
Jonathan Simone Bar-Eli
Sachin Mahashabde
Jeremy Suero

Follow this and additional works at: https://scholarship.shu.edu/petersheim-exposition

Part of the Databases and Information Systems Commons, and the Software Engineering Commons

Recommended Citation
https://scholarship.shu.edu/petersheim-exposition/89
The Team

Joshua Schappel
Project Lead - Developer

Jonathan Simone Bar-Eli
Full Stack developer

Sachin Mahashabde
Back-end developer and Machine Learning implementation

Jeremy Suero
Logo creation and database engineering, UI design
Application Overview

- Recommendation system for ASD-related articles
- Single page application (SPA)
- Hope to enrich the public’s knowledge about Autism
- Hope to enhance research ability within the field
The Layered Architecture Approach

- 4 main layers to the application
  - Controller
  - Business
  - Data Access
  - Common

- Layers are loosely coupled
  - Allows for easy updating
  - Allows to swap layers or create new layers with ease

- Scalability, to an extent
- Readability
Technical Stack

Python

Java with Spring MVC

Javascript, ReactJS

JSX, CSS

Back End

Front End
• Built in Java using the Spring MVC
• Each service within the business layer extends an interface
  ○ Scalable
  ○ New layers easily buildable
• Aspect-oriented programming (AOP) to connect all layers
• Jackson framework for serialization and deserialization of JSON files
• OOP design best practices
  ○ Favoring composition over inheritance whenever possible
• Design Patterns
  ○ Factory pattern
  ○ Visitor pattern
  ○ Adapter pattern
Machine Learning

- Machine learning is accomplished using a periodically updating python script
  - Data is pulled from APIs, then accessed by Python
  - TFIDF vectorization with Cosine scoring generates recommendations for each article
  - Results of recommendations are returned to the back end to display to user
Testing with Mockito

- Test driven development was a core design principle
- Each layer uses stubbing and object mocking to allow for unit testing across layer
-Mocks are injected into each layer to allow testing without disrupting the data
Application Demo
Concluding Remarks/Future Work

- Java gave the team the ability to fully realize the layered architecture approach
- However, on review, the JVM is not utilized to its full potential
- Other languages excel at I/O bound tasks, which could have provided us another option