This content-based recommender system takes a document corpus and reads in the text. It then creates and inverted index by entering the terms, as well as a list of documents containing terms and the frequency of the documents containing the terms. The system also creates a document by term matrix for later recommendations based on the normalized weights contained in the inverted index. It uses Apache’s OpenNLP to tokenize the terms, then porter’s stemming algorithm to stem the terms. It computes the tfidf values for weights and then normalizes them by taking the tfidf weights and dividing it by the sum of the weights times the inverse document frequency. It then uses k-means clustering to cluster the terms in the documents to group the terms contained into documents into subject groups. It uses the silhouette coefficient to measure the cohesion and separation of the terms is the documents (displayed in silhouette.txt). Though the silhouette values were not good, it did favor lower clustering, but the clustering was taking a very long time with smaller values. The system uses 11 clusters of which some were absent of terms when the clustering converged. The system uses the silhouette values with the highest value to determine the main subject of the clusters. For each term in the inverted index, the system uses the minimum value between that term and another to determine what term would be of most value to the user if interested in the subject of the cluster determined by the term with the highest silhouette value.

The system starts by checking if the silhouette values have been created, due to the fact that the calculations of the silhouette values took hours, it is detected and skips calculating those values and reads in a text file of the previously determined clusters based on those values. If silhouette.txt does not exist, the system will then re-create the clusters and calculate the values. If the file is present, it reads the documents, creates the inverted index, calculates the weights as well as the normalized weight. It imports the silhouette and associated values and assigns those values to the terms in the inverted index from the terms.txt file. It reads in the previously created profiles from a text file to keep the profiles of the users to add and enhance the user’s recommendations. The system asks the user to log in and checks the user profiles for name and password. If a new profile, the system asks the user to make a profile with username and password.

It then presents the user with the subjects determined by the silhouette values for each cluster and asks the user to enter the terms they are interested in. The system then presents the user with options to add terms, get documents, get associated documents or exit the system. The terms recommended are based on the closest term in the cluster based on the lowest cosine similarity difference between the term chosen and the other terms in the cluster. The user can add those terms to the list of terms they are interested in. The system allows you to get documents that contain the terms the user is interested in. The system then allows the user to read the text of the documents and decide if the document was useful to the user. If the user determines that the document is useful, it adds the documents to a list of rated documents. If the user wants to get more documents that are similar to the documents the user deems useful, the system calculates the cosine similarity between the rated documents and the rest of the documents in the corpus. The system adds those ratings and the associated documents into a sorted tree map based on the cosine similarity values and returns the top five documents. It then allows the user to read those documents and rate those as well. The system allows the user to do these actions until the user decides to exit the system.