Luke Staib: U39863533 2-8-21

EC500 Professor Osama

Homework 2 Phase 1: Operations for the Secure File Uploader/Ingest, Text NLP Analysis, and Newsfeed Ingest

Secure File Uploader/Ingest

- API Type: Procedure-based
- Operations:
    - UploadFiles(userID, files[], filetype)
        - Will upload file to account determined by user's ID, method of upload will change based on type of being being uploaded
        - For multiple files, number of files in file array will determine how many times this function is ran in a loop
        - This operation needs to be secure
    - DisplayStatus()
        - Will return progress of which files have been successfully uploaded, any errors, check return status of UploadFiles()
    - CancelUpload()
        - Will cancel current upload process
    - Leave()
        - Will go back to main hub
    - FileDelete(file)
        - Will delete file from current list displayed (in array files[])
    - FileEditName(file)
        - Will allow user to change name of selected file
    - OrganizeFileList(files[], organize_type)
        - Will organize files[] array with several different options
            - Alphabetical, Latest Uploaded, Earliest Uploaded, etc.
- Data:

  - Text Data: TXT, PPT, PPTX, RTF, PDF, DOC, DOCX, CSV, XLS, XLSM, XLSX, etc.
  - Zipped Data: ZIP, RAR, TAR, 7Z, etc.
  - Will handle large files, multiple files at once, etc.
  - No multimedia data (maybe?)
- Status:
  - Menu before upload
    - Option to upload one file or many files to a user's account
    - Option to leave and go back to the previous menu/main hub
  - During upload
    - Shows status of upload, displays any errors, shows how many uploads have been uploaded so far
    - Option to cancel uploading mid-upload
  - After upload
    - Display message to user that upload of files has been completed successfully, or return appropriate error
  - Files uploaded display
    - Will display files that have been uploaded, currently on user's account
    - Will be able to organize, delete, edit name of files

Text NLP Analysis

- API Type: Procedure-based
- Operations:
  - CreateKeywords(text or files[])
    - Takes in input from user and analyzes it using web-based data such as API data from many different news websites, generate keywords
  - ObtainArticles(keywords)
    - Keywords used from AssessData() will be used to obtain articles that discuss the topic that the text is discussing, obtaining relevant writing
  - AssessData(article_data, keywords, text or files[])

    - This is where the program will take all aspects, article data from ObtainArticles(), keywords from CreateKeywords(), and the original text or file/s to create a sentiment
  - SaveSentiment(userID, sentiment)
    - Save the sentiment that the program generates to a user’s account
  - EditSentiment(sentiment)
    - This is where a user can edit a sentiment that the program generated
  - DeleteSentiment(sentiment)
    - A user can also delete a sentiment and try again with different parameters
  - Translate(text or files[])
    - Translates text or sentiment into different languages
      - Will start with popular languages: English, Spanish, Chinese, French, Hindi, Portuguese, Russian, Japanese, German, Danish, etc.
- Data:
  - Text Data: TXT, PPT, PPTX, RTF, PDF, DOC, DOCX, CSV, XLS, XLSM, XLSX, etc.
  - Zipped Data: ZIP, RAR, TAR, 7Z, etc.
  - Extract data from other sources: Tweets, emails, reviews, social media posts, etc.
  - Data of files uploaded
- Status:
  - Text-input menu:
    - User can enter text or use uploaded files
    - Menu will show progress/status of text analysis
    - Program will show user sentiment that the system calculates based on the text inputted, user is then able to save, edit, or delete report generated

Newsfeed Ingest

- API Type: Procedure-based
- Operations:
    - DiscoverContent(search_text)
        - Takes in text that user enters to search for relevant information on that topic, uses keywords in the text entered
        - Uses NLP analysis to generate keywords and sentiments to return specific articles, see NLP analysis for those functions
    - OrganizeContent(organize_type)
        - Able to organize content that the program returns in different ways
            - Alphabetical, Most Recent, Most Viewed or Accredited(?)
    - ReadLater(userID, articleID)
        - Allows a user to save a specific article to their user account to read later
    - Leave()
        - Will go back to main hub
- Data:
    - Enter text based data into a search field, no files
    - Returns a list of articles that a user can browse through or save to read later
- Status:
    - Shows input field before user has searched anything
    - When user has searched, program shows progress of search, and then articles appear on the screen
    - User is able to organize articles with GUI, user is able save articles with indicator that the article has been saved, user is able to leave this screen