Contributions

Christopher Suh's Contributions:

- Worked with Jonathan on the XML Parser which included:
    - Making sure the parser returned the correct data in CSV Format
    - Parsed all layout files correctly and efficiently
    - Wrote method to store xml files parsed as strings
- Wrote a shell script to unpack tar files
- Parsed the data into CSV files
- Wrote documentation (on the wiki) for setting up everything on Mac
- Worked on the front-end, specifically:
    - Cleaning up the stylesheets
    - Styling the webpage with bootstrap
    - Fixing the navbar
    - Styling various items such as buttons, inputs, etc.
    - Created the About page, including images of each team member
- Produced application screenshots for matching applications

Jonathan Lin's Contributions:

- Parser
    - Created apps package that organized structure into Application/File/Component objects
    - Wrote method to search through manifest/values to find and store application names
    - Wrote code to file walk through all app folders, then layout and values folders inside those app folders, then all xml files in those layout values folders
    - Wrote visitComponent method which goes through components of XML files recursively so parent child hierarchy could be stored, also wrote the previous iterative version
    - Wrote methods (findDir, findFile) to find specific files/folders in the app folders
    - Used XML DOM Parser to gather relevant information from the XML layout files found
    - Wrote code to store attributes in hash maps for component objects
    - Wrote code to count total # of attributes to assess which ones are most popular
    - Constantly modified data storage structure/csv print output to correspond with database layout needed for django/search algorithms
    - Wrote method (printCSV) to output XML data to csv
- Frontend/Other
    - Ran search engine/apps to generate screenshots for test cases in archives folder
    - Created javascript function to format and display xml string input on front end

    - Modified mysql database settings so XML data could be inserted properly (sizing, variable types, ordering)

Michelle Chen's Contributions:

- Worked with Ann to create/plan/remake the database schema and database
- Setup Django
    - Connected frontend to backend
    - Created the models in models.py
    - Configured settings in settings.py
    - Configured initial urls to redirect pages in urls.py
- Configured Django to work with Haystack and Solr
- Setup Solr
    - Built solr schemas using Django's built in command build_solr_schema
    - Setup indexing functions to store data from the database in indexes and store complex data that makes searching faster (stored in search_indexes.py)
- Added some bootstrap to front-end
    - Created css files to help separate the style and the html
- Created 4 search algorithms (component, cosine similarity, hamming distance, and euclidean distance)
    - Optimized search algorithms as much as possible
- Rewrote 4 search algorithms to handle multi-file input and switched from file-centric results to application-centric results
- Linked javascript to the html front end to enable pagination that helps break up results
- Created the form to allow users to input new applications and files associated with that application
- Created the backend to handle the new application and file data and input it into the database
    - Also enabled solr reindexing to index the new files
- Redirected incorrect user input to an invalid file html page
- Returned counts of components to the results page
- Wrote documentation on the gitlab to get things set up in linux, an explanation of the directory structure in Django, and Solr setup instructions
- Worked on parts of frontend here and there
    - Reformatted the output from results to show the components of input files

Daniel Powers' Contributions:

- Worked on Front-end throughout the project
- Initial design of front-end in sprint one, and continuously changed and improved front-end design.
- Built results screen, home screen, invalid input screen
    - Added ability to send multiple files to parser

- Tested for and handled invalid input
- Made display for results, including popups for XML files, and display statistics from each file in the search results
- Edited and cleaned up html and css for efficiency and better design.
- Writing testing instructions in readme and usage page on the website
- During first sprint, organized gitlab including:
    - Writing user stories, tasks, and sprint goals
    - Making issues from tasks, assigning them and making tags

Ann Gao's Contributions:

- Worked on database design throughout project
    - Changed design according to data needs
    - Created ER models for each database design
    - Created SQL files for each database implementation
- Kept Gitlab up to date with issues
- Added all data from CSV files to database
- Wrote parsing algorithm to parse user uploaded files for Component search
- Wrote parsing algorithm to parse user uploaded files for Hamming/Euclidean search
- Worked on front-end
    - Connected drop-down menu to different search algorithms using JavaScript
    - Styled different iterations of front-end design
    - Made navbar
    - Integrated uniform design across pages
    - Cleaned up code