

Search'M (Search and Map)

Search'M is an application that allows users to search through specified RSS feeds, documents, and images and can view results in Google Earth. The user can specify individual folders, files, RSS feeds, or drives that they want to be indexed. RSS feeds will only include articles that are within a window of time which is specified by the user. After items are indexed, the user has the opportunity to manually label an item (keywords, location, category, ect.) through Search'M. Locations can be assigned by city and state or by longitude and latitude. There will be predefined categories, such as news, travel, and work, and will allow users to create their own categories. Once there are indexed items, the user can now search based on keywords, locations, dates, categories, and/or the type of an item. The results will be displayed in Search'M and each result will include a link to the item, date, location, type of the file, category, thumbnails, and an excerpt of text. A KML (Keyhole Markup Language) file will be generated for items that have geographic data which will be viewed in Google Earth. For each result with geographic data, a placemarks is displayed with its corresponding result data. When a placemark is hovered over, lines will be pop-up to show the relationship between that item and other items on the globe (similar keywords, categories, dates, ect.). A feature that may be included is the ability to search for items based on the bounding box (the top right, top left, bottom right, and bottom left coordinates) that are being displayed in Google Earth at the time of the search. The bounding box in Google Earth will be the search parameter and the result will be any item that is within that bounding box.

The first part of the final demo will show that Search'M can index different file types as well as RSS feeds and that uses can manually label items as well. Afterwards I will show how the search engine works and show that the application only indexes RSS articles that are within the specified time period as well as showing that the information for each result is correct and the links to the item works. I will then delete files that were indexed, search for those files again, and show that the files are not in the system anymore. I will also demonstrate the search filters and how we are able to search by multiple filters. I will then search for files that I know have location data and display the resulting KML file in Google Earth, explaining the information shown for each placemark and presenting the lines that popup when hovered over and what they represent. Finally, I will search for files that I know do not have any location data and show that the KML file is empty. Furthermore, I will also present any other small features such as displaying all of the indexed items in Google Earth and the feature for searching by the current bounding box in Google Earth.

Related products include Google Desktop¹ and Mac Spotlight² that are personal desktop management software, Flickr³ and Yelp⁴ that displays results with location data in maps, and Delicious⁵ and other RSS aggregators. My product would combine the three concepts; cataloging, geotagging, and aggregating into a search application that can work with multiple data types.

I plan to use Java as the language for this project and I have chosen to use Oracle Berkeley Database Java Edition as the embedded database. I have not used Berkeley DB before but I have worked with SQLite and I would like to use a Java embedded database for this project. For displaying placemarks and data in Google Earth the Google KML (Keyhole Markup Language) API will be used as well as JDOM, a Java library that simplifies reading and writing XML files to create the KML files. To extract image metadata to check if an image has location data, I will be using the Java Image I/O API. For reading and parsing RSS feeds, I will be using the ROME v1.0 library which is a set of Atom/RSS Java utilities for parsing and generating simple syndications as well as handling different RSS formats. I am familiar and very comfortable with the KML API and the Java Image I/O API. I have not specifically used JDOM or ROME but I have used similar APIs in C# and after researching these APIs I do not think there will be an issue.

The main components of my system is indexing, labeling, searching, and displaying in Google Earth. They fit together as a natural progression and the final component of seeing the results in Google Earth is just another way to visualize the results. I will be building the cataloging, algorithm for determining location data and keywords, file system monitor, data bundles that store all the information of each item, user interface, and the search algorithm. The APIs that I am using provide the tools I need for parsing different file types and allow me to represent my results in Google Earth.

There are a few algorithmic challenges in this project. Monitoring the files that have been indexed is an algorithmic challenge because the files can be moved, deleted, or corrupted, since the time it was indexed. A solution would be to use file monitors that check the state of the files or folders to see if any changes have happened as long as Search'M is running. Also, the application must make sure that the buffer containing the indexed items has not reached its memory limit and if it has it must remove files based on the date it was indexed. Finding key words in the text, determining relationships between indexed items, and finding location references in the text are going to be very difficult tasks. I am still in the process of thinking out solutions for them.

The next steps in advancing my project would be to examine the APIs I have chosen and make sure that they accomplish what I need them to do, look for other tools that may be beneficial, list out and elaborate on every functional requirement, create use cases, design the structure of the system, and create a timeline for the project. My project is going to win the Best Senior Design Award because it provides a new and interesting way to view search results of personal files and RSS feeds. It will allow users to see the geospatial relationship between news RSS feeds, work related files, and their images, as well as having a desktop search that allows users to customize and label the indexed items. Furthermore, it allows the user to creatively view the world as an interconnected web of events. In the 21st century where globalization is at its peak, this program allows users to see events that happen around the world.

[1] <http://desktop.google.com> [2] <http://support.apple.com/kb/HT2531> [3] <http://www.flickr.com>
[4] <http://www.yelp.com> [5] <http://www.delicious.com>