Sherwood, Joshua; Gumirov, William.

// functional vs non-functional requirements: 5 func, 2 non-funct for project

Why is this an interesting project?

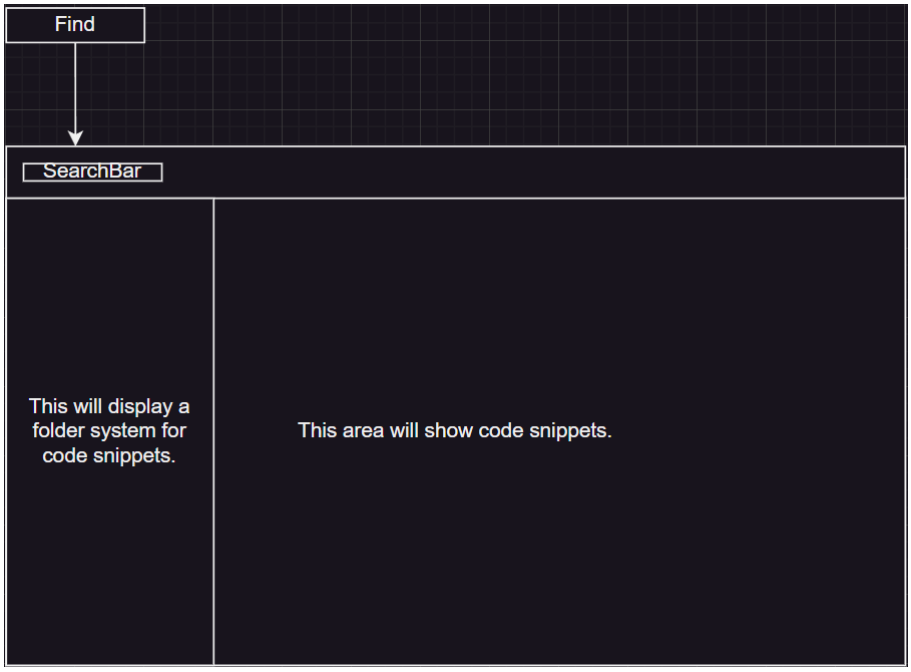
- Building on previous ideas is one of the ways science, including computer science, has advanced so effectively. This idea allows you to easily search for previously written code snippets that you have labeled, ranging from just a few lines to the entire file, which is interesting to us because it would be useful not just to us but other developers outside GW, and we could build on it once the class finishes to make it even more useful. The usefulness and scalability aspects make it an interesting endeavour.

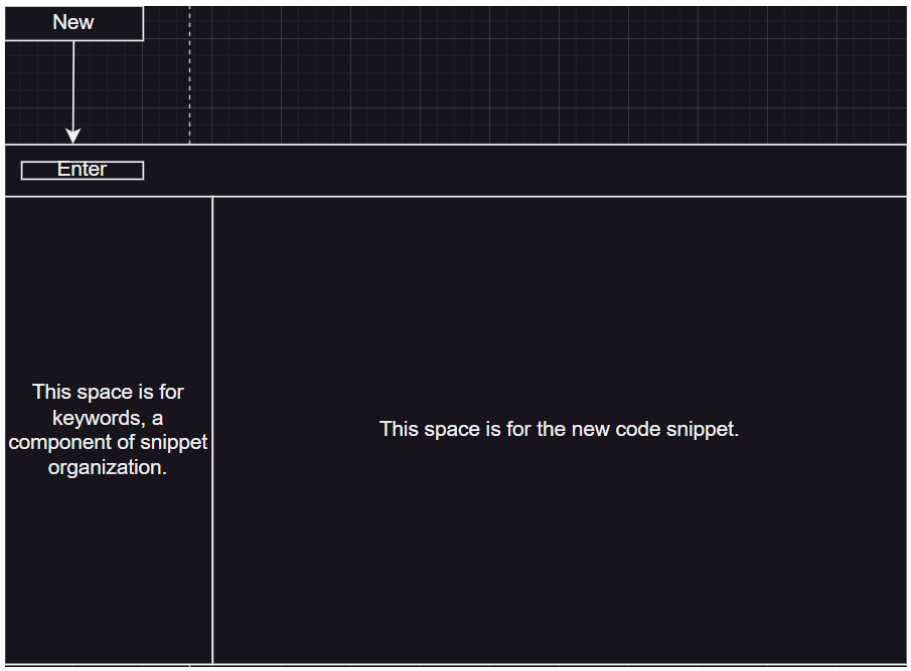
What problem are we trying to solve?

- We want to make it easier for people to integrate previously written code into their projects by allowing them to label snippets of code with chosen keywords and then search for them with these keywords. Multiple queries will appear at first (if there have been a large number of labels), but these should disappear as the keywords become more specific.

GUI

- The GUI of this program will be a separate window which may be accessed by a user. Within this window, we will allow users to search for & organize code snippets, similar to how files are organized on a computer system via folders. We have decided not to use JavaScript because the functionality it would offer would probably be exceeded by creating more points of failure. However, if we have time and think this might work, we might re-consult with you about this. But for now, the plan is to have a two-button GUI pop up when you open the app, with NEW and FIND, their functionality based on the name. For new, you would copy your code from your IDE or code editor into our app, provide the keywords, and it would be added to our database. For searching, you enter your keywords and receive the best results for code snippets that match these keywords.





Networking/Multi-threading

- Inserting a new snippet into our database or finding an existing one will put strain on the GUI's single thread (if we use Swing or JavaFX, which is likely). This might cause our window to slow down or freeze, which would not be desirable. So to avoid this, we will use multi-threading.

How is this project challenging?

- It requires implementing and optimizing database and search functionality in a GUI. In class this and last semester, we have learned each of these elements apart, and this project combines them together. The optimization with threads will add to the challenge. If this doesn't sound challenging enough, we are willing to implement a server aspect to this for collaboration, just let us know.

Testing plan & deliverables

- The final result will be a GUI that allows you to search for code snippets that you have stored with keywords, as well as navigate through a folder system you have designed, and we have provided the interface for, to organize them. To test, we will make sure the

database is properly receiving the new snippets. This can easily be confirmed through accessing the data stored within the database. Once this has been tested, we can move on to testing the search functionality: that it returns the desired results for the keywords provided. We will start with easy cases where keywords are distinct to ensure basic functionality. Then, we will handle the case where there are multiple results for the same key words. Code snippet autocomplete functionality will be tested after this. Finally, the effect that these operations have on our GUI will be tested by inputting an increasingly large number of snippets, including very large snippets (we will have a size limit on these that has not been decided yet), and then retrieving them. We will make sure that our GUI runs as smoothly as possible even during these cases, and doesn't crash, by using multi-threading. Essentially, we will test the case where the max size and max input rate are inputted and retrieved by the testers.

Manager specifics

- The Team Manager (TM) shall have reduced responsibility for actual coding - for example, other team members will observe office hours to ask questions and solve problems of implementation.