**FURPS+**

Functionality:

* Allow the user to input an executable file and receive back an approximation of what the source code of the executable looks like
* Optional: allow the user to further train the model on their own data
* Optional: a feature that will detect what compiler was used to create the executable

Usability:

* The user should be able to use the program completely from the command line
* The program will prompt the user to supply parameters for the program, locations of the executables, etc.
* The program will output the code prediction on the commandline and save it to a file if the user requests to.
* Optional: a more user friendly command line user interface (something like the metasploit framework).
* Optional: a GUI that allows drag and drop and other graphical ways to open files and display the source code

Reliability:

* The application will be available all of the time because all of the resources needed for the application will be local to the user’s machine
* When training large amounts of data, the program must ensure it isn’t killed by the operating system for running too long

Performance:

* An average modern desktop computer should be able to support the system
* Users training the machine learning model within the application will need more specialized hardware for the application to perform properly
* Better assessments of performance will occur when models have been trained to do benchmarking on

Supportability:

* Optional: When training data, show side-by-side comparisons of source code and generated source code, highlighting differences
* The code will be open source, allowing others to maintain the application

Physical requirements:

* Nvidia GPU (optional) - can be used to decrease training times compared to a cpu
* More processing power will result in faster training

Interface Requirements:

* The program will run on the command line. If it is written in python, it will be usable via the python interpreter

Implementation Requirements:

* Python may need to be installed on the client’s computer. Dependencies should be resolved during initialization

Design Constraints:

* Program must be able to continue running for long periods of time

**User Stories:**

1. Decompiling a file: The user, after starting the program, is prompted to pick an action. They enter decompile and provide the path to the executable, and the path to where they want to save the results. The program then indicates all of the parameters provided are valid and begins running the model. The results are displayed on the screen and saved to a text file at the user supplied location.
2. Decompiling a file via GUI (optional): The user clicks on the program icon, and is presented with a blank screen, with a file menu at the top. Double clicking the screen, or selecting “Open File” via the file menu will open up a file explorer screen. From there the user will be able to select there executable to decompile. Alternatively, the user can just drag a file onto the blank screen. After validating the program is an executable, it will attempt to decompile it into c code. The results will then be displayed on the screen.
3. User training (optional): The user has a directory of c code files that they want to train the model on. The user, after starting the program, is prompted to pick an action. They enter “train” and provide the path of the directory of c code and a path on where to save the finished model. The program verifies that the directory contains all valid c code, and a compatible c compiler is installed on the system. The program will start the training process and output updates from the training framework to the screen. After training, the program will output the results of training and testing the model.