Things that can be done:

* Look into Bluetooth support and add a working library from vcpkg that would support the library cross platform.
  + To my knowledge “simpleble” would work but research is needed.
  + Once a library is found that will work cross platform for C/C++ with the C++ code, write Connection\_Bluetooth .hpp and .cpp files where the .hpp file inherits from Connection.hpp
  + See Connection\_VCP.hpp and Connection\_HID.hpp for examples on how to do this
  + Next develop the methods of the Connection\_Bluetooth.cpp file using the library that was chosen.
  + Bluetooth has no additional API changes but some adjustments of the Bluetooth data may be necessary for the read call to function
  + The read call used in connection is expected to get the character amount that was given regardless of the length of the data actually given from the device.
  + This is because the read call is expected to act as the data is given as a stream.
  + This means that if the data is not given as a stream it must be made into a stream as done in Connection\_HID class
  + This is so the read call can be called across the library and different functions can be called or the function can be skipped in the case that it is not correct
  + The write call should expect the entire call to the Connection class in it’s entirety making it rather simple compared to the read call
  + Once Bluetooth support is added to the connection class a function must be made in Extension.h and Extension.cpp to allow users to use this new connection class in python and java and also be able to use it more easily (rather than using setConnection method or creating the graphiti object with a connection class which is limited to C and C++)
  + Both startUpBluetooth and shutDownBluetooth must be created in the extension. Use the startUpVCP and shutDownVCP as a reference
  + Once these have been tested, the can be added to the C wrapper, python wrapper, and java wrapper. Use the startUpVCP and shutDownVCP as a reference in the wrappers
* Add operating system changes to CMakeLists.txt files to support Linux and macOS
  + Add any additional changes to other files as necessary
  + All libraries used from vcpkg should be cross-platform as to not inhibit the Graphiti library as a whole from being cross-platform
* Create an exe file that will do the entire installation process of the library from GitHub and uninstall after leaving the necessary files for running in C, C++, and python
  + Most realistic for a low-level user
  + Could have one for java but would be more complicated
  + Exe file for each language
    - C
    - C++
    - Python
    - Java
  + This can be done because ps1 script files can be converted into exe files
* Test whether the current configurations are standalone as in they meet these criteria:
  + Do not require vcpkg to be installed on the computer and can be run minimally with the asio header file, hidapi header files, and hidapi.dll