SEAD Backend Testing Document

Interaction with the API

User Story: *As a GUI developer, I want to query the database for signals within a certain range*.

Actor	System
1. Initiate an HTTP GET request to the API, providing appropriate filters at GET parameters.	2. Parse GET request and build a query according the the specified filters. This query is executed and the results are formatted and rendered.
4. The client receives the raw data in a two dimensional Python array, with the first row being the header of the columns.	3. Results are sent to the user.

Interaction with the Landing Zone

Actor	System
1. SEAD plug requests configuration from server by sending its header containing its serial number and time.	2. Landing zone reads header, sends an ACK, constructs a configuration and returns it to the client. Landing zone sends an OKAY after it is finished sending the configuration.
3. SEAD plug receives packet(s) and begins to collect data. It sends it to the server when unknown criteria has been met.	4. Landing zone receives data and bulk inserts the data to the database. Landing zone sends ACK.
5. Client receives acknowledgement from the server.	

Test Coverage Summary

The landing zone and API have not been fully unit tested. Due to the short development period and ambitious scope of the project, we prioritized features over through testing.

In order to reduce the number of possible bugs, we used a simple compiled strictly and statically typed language for the landing zone and made all of our code as simple as possible to avoid hidden bugs. We also avoided objects and inheritance which can be sources of complexity and bugs. The single set of unit tests is present only to ensure accuracy of some conversion functions needed by the module which decodes packets from the SEAD plug.

Known issues

The subset API option does not always return the specified number of points even when doing so is possible. This is because we are selecting every n^{th} point where n is an integer. We may need to tweak our rounding code.