

## Service Toolkit Catalogue

### Introduction:

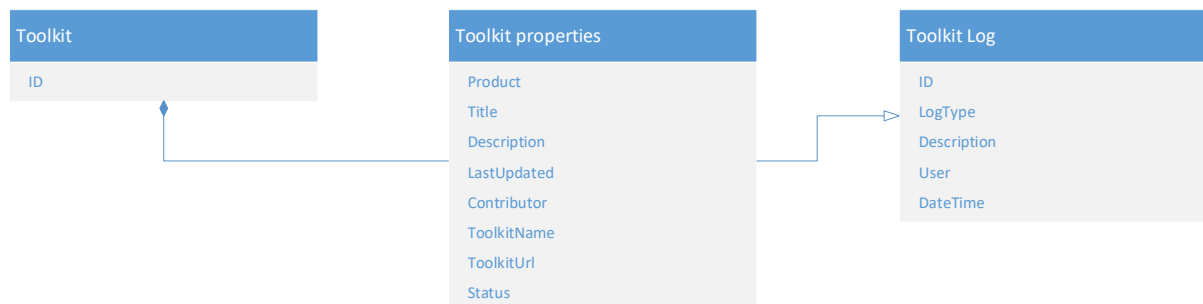
The purpose of the application is to be able to read a JSON file containing information about toolkits available to implementation consultants. The toolkits consist of modular solutions that have been developed to answer specific issues or challenges. The objective is to run an application that will read the JSON so the application can be used to perform several tasks against the data.

The application will allow a form application to be presented to a user. The application when executed will present the user with options to read data, insert a new object or modify an existing object. Each option will in turn execute a different form or function based on the operation to be executed.

The application will also record logs for insertion and modification actions and will present options to output or save the log locally into a pre-set file format.

### Data Structures:

The data structures used by the application will be for insertion, searching and modification. It will be a linear data structure and is an array of entities that use the same pattern in the definition of the objects and pairs. The application will be able to iterate through this array based on the access type (read, insert or modify) and the properties selected or set in the application by the user through a range of statements and functions. The data structure should consist of object types that contain a primary key value with associated values. In the toolkit case, the primary key for an object is a numerical reference value (32571). Any actions performed by the application should always be based on a key value.



*Example:*

```

{
  "Name": "SampleManagerToolkits",
  "Description": "Thermo Fisher Services Toolkits Catalogue",
  "Author": "Simon Bolder",
  "LogFile": "ASTLogFile",
  "LogFileDir": "LogFiles",
  "Toolkits": [
    {
      "32571": [
        {
          "Product": "SampleManager" },
        {
          "Title": "[AST] Sample Explorer Pivot Grid" },
        {
          "Description": "This AST helps displaying Sample, Test, and Result historical data into SampleManager." },
        {
          "LastUpdated": "4/28/2020" },
        {
          "Contributor": "Jerome Malessan" },
        {
          "ToolkitName": "[AST] Sample Explorer Pivot Grid V1.2.zip" },
        {
          "ToolkitURL": "https://support.thermoinformatics.com/kb/attachment.aspx?kb=0x0000000000007F3B&at=0x00000000000038411" },
        {
          "Status": "Active" }
      ],
      "32572": [
        {
          "Product": "SampleManager" },
        {
          "Title": "[AST] Contract Module" },
        {
          "Description": "This AST helps providing the ability to manage Contracts for Customers within SampleManager LIMS." },
        {
          "LastUpdated": "4/28/2020" },
        {
          "Contributor": "Javier Fraile" },
        {
          "ToolkitName": "[AST] Contract Module V1.3.zip" },
        {
          "ToolkitURL": "https://support.thermoinformatics.com/kb/attachment.aspx?kb=0x0000000000007F3C&at=0x00000000000038412" },
        {
          "Status": "Retired" }
      ]
    }
  ]
}

```

Fig1. JSON Example of linear data structure.

### Algorithms:

The application will use 3 algorithms. The algorithms are supported by flowcharts to outline the decision-making process with each action.

**Read/Search:** The application should allow the user to search/read data from the data structure. It should provide fields for key values to be entered/searched on to return the correct values.

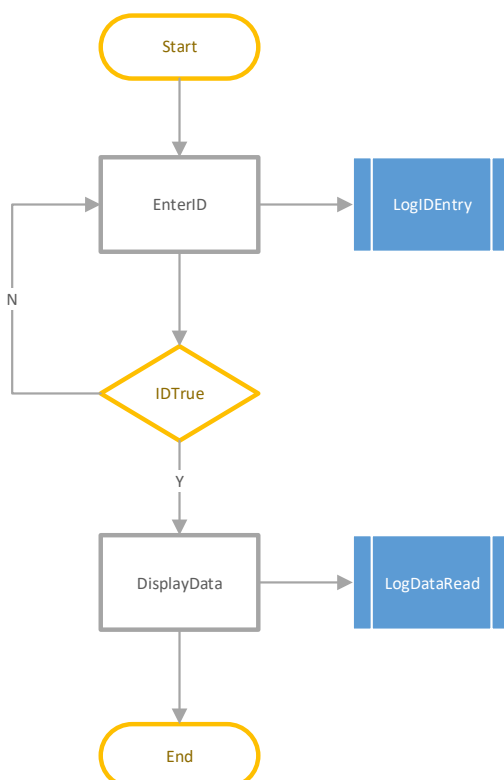
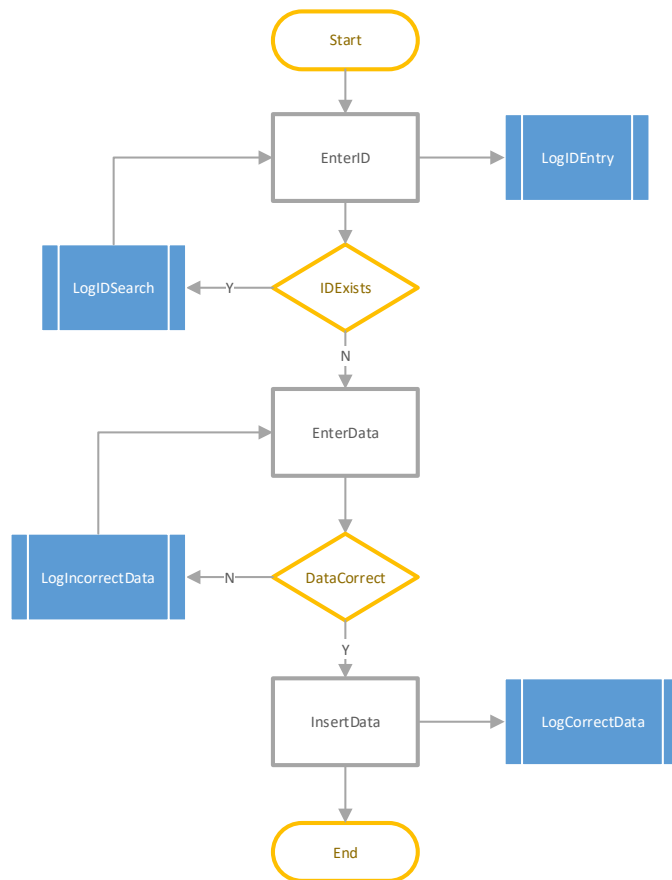


Fig2. Flow for reading data

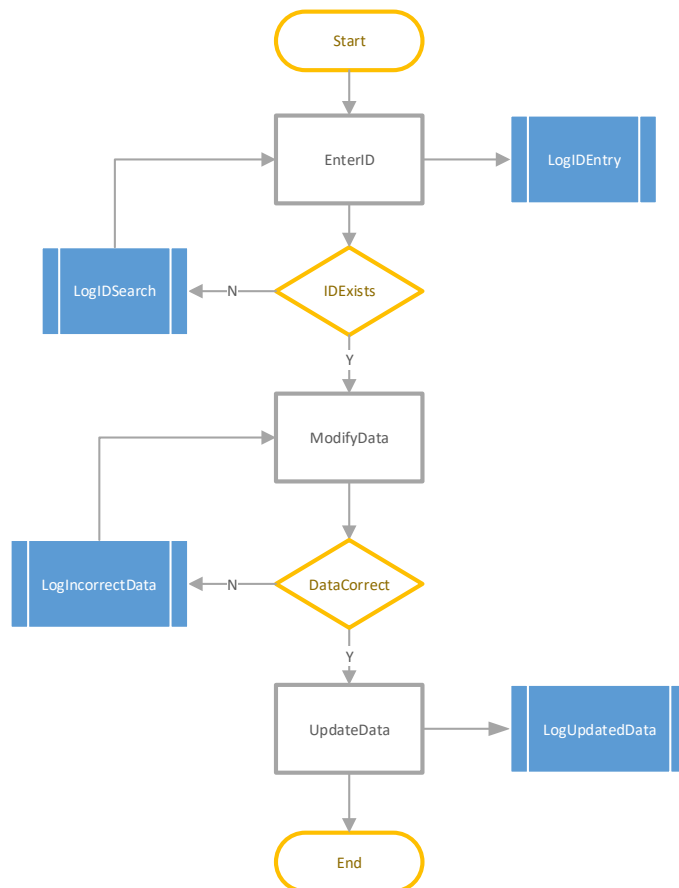
**Write:** The application should allow the user to write new data into the data structure. It should provide for certain mandatory fields or key values to be entered.

Should also provide error checking if the dataset is incomplete and that datatypes are correct.



*Fig3. Flow for inserting data*

**Modify:** The application should allow the user to modify existing data into the data structure. It should provide for certain mandatory fields or key values to be entered to return the correct values. Should also provide error checking that dataset is correct and datatypes are correct.



*Fig4. Flow for updating/modifying a record*

### **Test Plan:**

Testing should cover scenarios for insertion, search and modification as well as logging. Based on the flow diagrams the following sections will define the points for testing. The overall test plan will be validated with an RTM (Requirements Trace Matrix) that will define the functional requirements and be categorised into areas - insert, read, modify). A test plan document will also be raised referencing the RTM items to provide evidence (screenshots, textual summary) of results. The test scripts will be written in the context of a user.

### **Example of a test**

1. *Select Insert record*
2. *Set a numeric value for the ID*
3. *Set text value for a description*

Insertion – Testing will be performed to cover the following

1. Insert a new record
2. Verify new entry by logging in and selecting the search option
3. Check data in the data structure corresponds with the data entry
4. Verify log file has been updated with the insertion

Read – Testing will be performed to cover the following

1. Read a record
2. Check data in the data structure is what is displayed in the form

Modify – Testing will be performed to cover the following

1. Modify a record
2. Verify the modified entry by logging in and selecting the search option
3. Check data in the data structure corresponds with the data entry
4. Verify log file has been updated with modification

Expected test results will be a yes or no to the sequential test list and screenshots of each test scenario for data verification steps.

The application will not use 3<sup>rd</sup> party libraries but will require the import of standard Python libraries to allow the programming and rendering of a user form. The program will be using the text box and label controls to display data and will consist of functions to perform the relevant steps. The program will use a variety of loops and conditions to apply logic to the steps.