SAND2015-0799R

Acceptance Equipment System Data Acquisition and Processing Utility

Rowan Fakhro

I. Introduction

My internship at Sandia National Laboratories took place in the Department of Sensors and Embedded Systems, which is tasked with, among many things, the non-destructive testing of thermal batteries. The Acceptance Equipment System (AES) is a flexible rack system designed to electrically test thermal batteries individually for internal defects before they are stored in the battery stock pile. Aside from individual testing, data acquired by the AES is used for many things including trending and catching outliers within the tolerance levels of a particular battery type, allowing for the development of more refined acceptance requirements and testing procedures.

II. Task and Requirements

All measurements and test environment information are stored in a database on the AES, but being a recently designed system, all data processing and display has been done manually, costing a great deal of time and money on a task that can be automated. As such, after familiarizing myself with the system, operating procedures, and data analysis methods, my task was to use Python to develop a data acquisition and processing utility that can automate the generation of product reports. In addition, I needed to thoroughly document my work. There are three users that need potential accommodation with this utility: an operator, a developer, and a maintenance user. My task applies to the operator use case, which has the lowest level of privileges. The operator use case must be straight forward with minimal input required from the user. The deliverable of this task is an executable that can function without having to download Python and the corresponding packages in addition to my documentation.

III. Approach

In approaching this task, I divided the problem into six obstacles: (1) learning the Python language, (2) learning how to connect with the database and perform the necessary queries, (3) learning how to program the graphical user interface, (4) learning how to present and format data in a plot, (5) learning how to organize data into CSV files, and (6) learning how to organize information in a PDF file.

IV. Behavioral Diagrams

The data acquisition and processing utility can be modeled through the following use case:

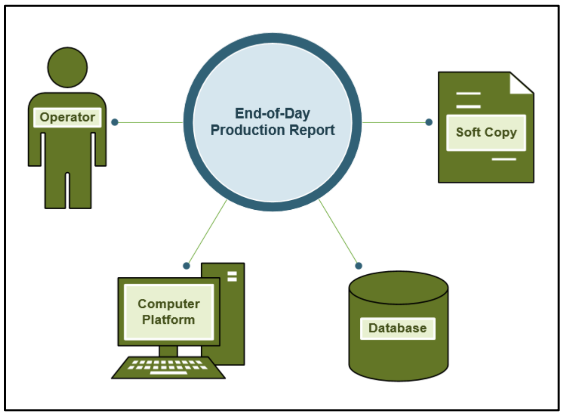


Figure 1: Use case diagram of production report utility

As can be seen in Figure 1, there are four elements that play key roles in this use case: (1) the user which interacts with the computer platform, (2) the computer platform which runs the utility that will access and process the data and present them in a structured format, (3) the database which stores all of the data, and (4) the soft copy which is the product report generated at the end of the process for data analysis. An overarching view can be represented with the following activity diagram:

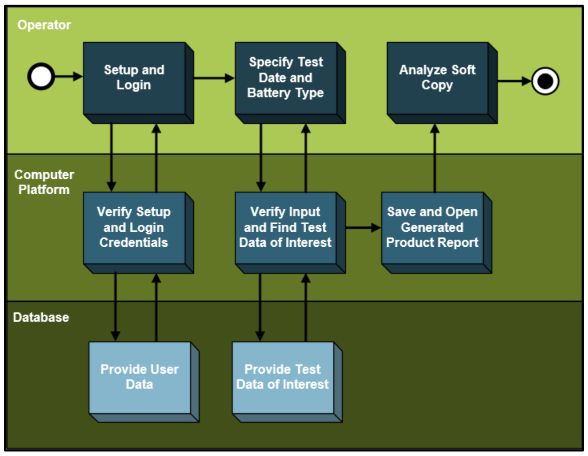

Operator
Setup and Login
Specify Test Date and Battery Type
Analyze Soft Copy
Computer Platform
Verify Setup and Login Credentials
Verify Input and Find Test Data of Interest
Save and Open Generated Product Report
Database
Provide User Data
Provide Test Data of Interest

Figure 2: Overarching activity diagram

In the activity diagram in Figure 2, the interface between the operator and the computer platform can be represented by the graphical user interface (GUI), while the interface between the computer platform and the database can be represented by the MySQLdb application programming interface (API) which allows for the connection with the database and the execution of queries. Note that in Figure 2, each action is identified with a different shade of blue depending on whether it is an action performed by the operator, the computer platform, or the database. This color-coding will be consistent throughout the report.

i. Initialization

On the execution of the utility, the following login GUI will appear:

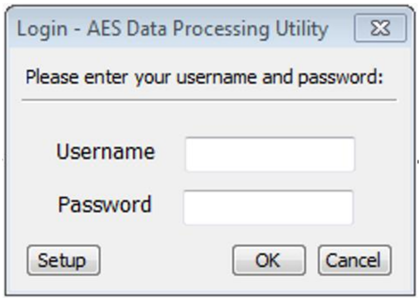
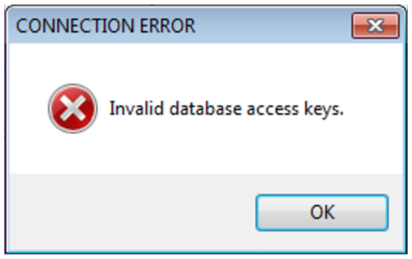

Login - AES Data Processing Utility
Please enter your username and password:
Username
Password
Setup
OK
Cancel

Figure 3: Login dialog

Before logging in, if this is the first time the utility is used, the user must set up the database access keys and specify the directory in which they want to save their product reports. If the database keys are not provided or they are currently invalid, the following dialog will appear:


CONNECTION ERROR

Invalid database access keys.

Figure 4: Database connection error dialog

After pressing the "Setup" button, the following GUI will appear:

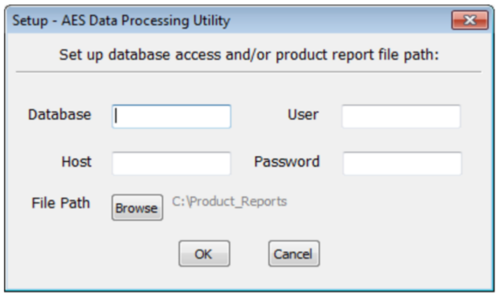

Setup - AES Data Processing Utility
Set up database access and/or product report file path:
Database
User
Host
Password
File Path
Browse
C:\Product_Reports
OK
Cancel

Figure 5: Setup dialog

The connection error dialog in Figure 4 will always be followed by the setup dialog in Figure 5 to allow for an immediate update of setup parameters.

In addition to allowing for the input of the database keys, the setup dialog allows the user to browse directories to specify the directory in which they want the generated product reports to be saved. The current directory will always be specified next to the "Browse" button, but if the path is too long it will be abridged to fit on the dialog. After pressing "OK," a dialog will appear to inform the user that the setup was successful and of which set of parameters were updated.

After the setup, the user can now attempt to login. Upon selecting "OK" on the login dialog, the computer platform will check the login credentials to see if they are present in the database as modeled by the following behavioral diagram:

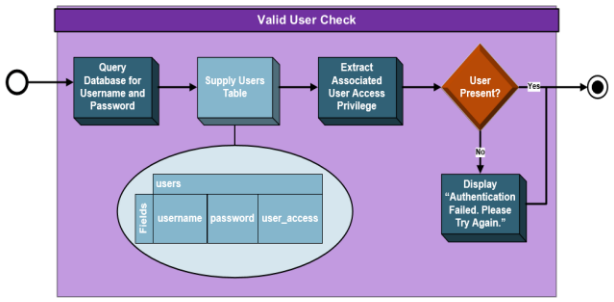
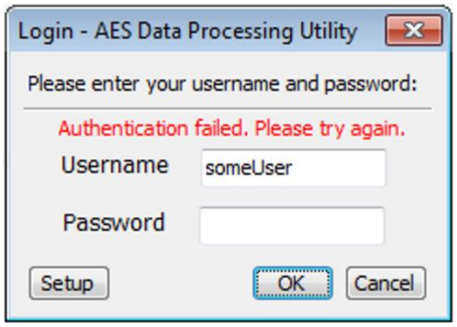

Valid User Check
Query Database for Username and Password
Supply Users Table
Extract Associated User Access Privilege
User Present?
Yes
No
Display "Authentication Failed. Please Try Again."
users
Fields
username
password
user_access

Figure 6: Checking validity of username/password input

As can be seen in Figure 6, the computer platform will query the database for the username and password and extract the associated user access privileges, which indicates whether the user is an operator, a developer, or a maintenance user. In the case where the user is not present or the input is invalid, a message will appear on the login dialog as in the following:


Login - AES Data Processing Utility

Please enter your username and password:

Authentication failed. Please try again.

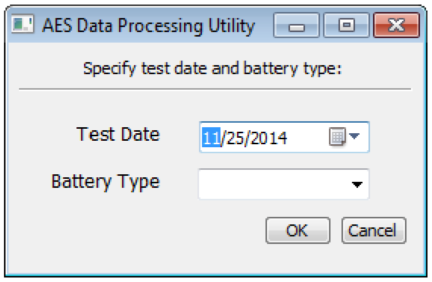
Username: someUser

Password:

Figure 7: Invalid username/password input

ii. Test Specification

Upon successfully logging in, the AES Data Processing Utility will appear:


AES Data Processing Utility

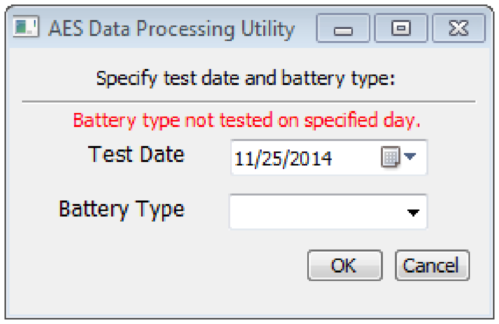
Specify test date and battery type:

Test Date: 11/25/2014

Battery Type:

Figure 8: Data processing utility

As can be seen in Figure 8, the GUI will take in a test date and a battery type which can be selected through the drop button, in which case the date can be selected through a calendar and the battery type can be selected through a drop list, or the user can input the parameters manually. The computer platform will then take the input and check to see that the specified battery type was in fact tested on the given day. If it was not, the GUI will display a message such as the following:
After validating the input, the computer platform


AES Data Processing Utility

Specify test date and battery type:

Battery type not tested on specified day.

Test Date: 11/25/2014

Battery Type:

Figure 9: Invalid battery type and test date combination

will then query the database for the data necessary to construct the product report.

iii. Product Report and Organization

The product report generated consists of a PDF which contains the information regarding the general test setting in addition to all tests run specific to the particular battery type specified as well as plots of that data for the specified day. Since these production reports are on a need-to-know basis, a sample will not be included. The utility will only plot the data of tests that passed, and will also put that data into a CSV file to accompany the PDF. In the case where there are different types of tests done for that battery type, there may be multiple CSV files generated, each corresponding to a different test.

The PDF and CSV files that make up the product report will all be saved within the directory specified in the setup dialog. Within this directory, the product reports will be organized such that there will be a folder for each product report generated which will be titled according to the date of the test(s) and the model number of the batteries tested. Each folder will contain a single PDF of the product report and the accompanying CSV file(s) if present. In the case where the operator tries to generate a product report more than once, a dialog will pop up asking if the user wants to replace the previous product report generated. If not, the directory name will be altered with a number at the end of it to allow the user to distinguish between directories for the same date/battery type product report.

The structure can be seen in the following figure:

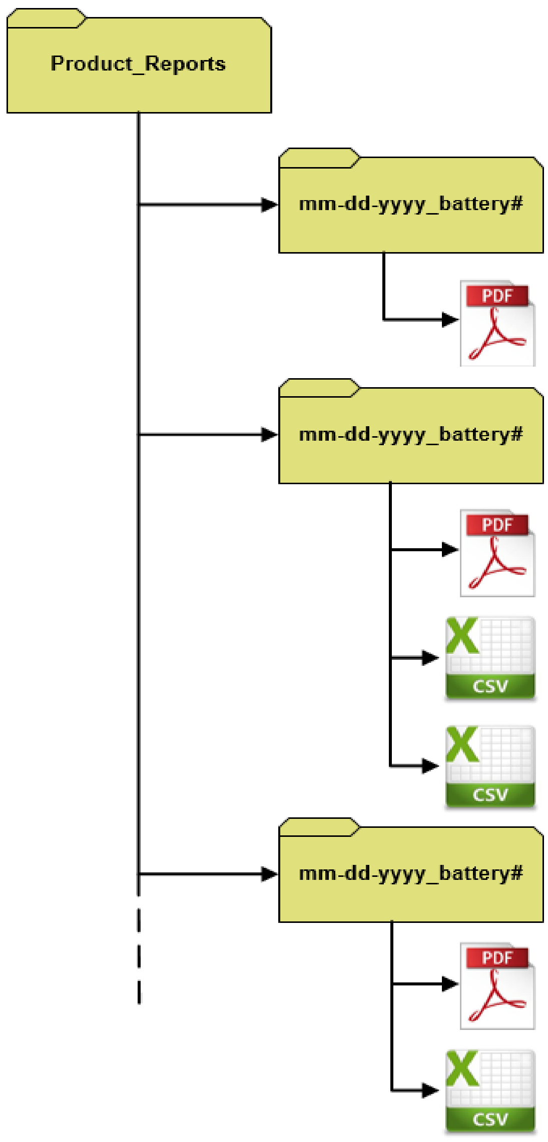


Figure 10: Directory structure

The names of the folders in Figure 10 are made using the user input of the date and the battery model number. For example, if a product report was generated for November 25, 2014 of the battery model BAT555, the directory it will be saved under will be “11-25-2014_BAT555” and it will be inside the directory specified in the setup dialog. If this same report was generated a second time without replacing the first one, the directory will be “11-25-2014_BAT555_(1).”

V. Software

The Python packages used, as applicable to this utility, are the following:

- MySQLdb – provides the ability to interface with the database and perform queries
- wxPython – provides the ability to program GUIs that are native to the operating system
- Matplotlib – provides the ability to generate and format plots
- Numpy – provides data types for data handling
- Reportlab – provides the ability to generate PDFs
- Py2exe – Provides the ability to create executables from Python scripts

The modules used in the standard Python library are the following:

- os – provides the ability to navigate and change directories
- subprocess– provides the ability to open files
- csv – provides the ability to generate csv files
- math – provides mathematical functions
- cStringIO – allows for the representation of different data types as a cString

In addition to Python, Microsoft Visio 2013 was used to generate the behavioral diagrams as well as the directory structure for program documentation.

VI. Conclusion

From the utility described above, it can be seen that the requirements that the utility must generate a product report with minimal input necessary at the user end has been met.

Further development would be to add a developer and maintenance use case which would accommodate those users. Since these users have higher access privileges than an operator, they must also have the ability to generate product reports like the operator.