



DataSwitch Data Sweeper (DS)²

by Joseph H. Farrar and James (Lee) Whitlow

INTRODUCTION

Purpose. The purpose of this Coastal and Hydraulics Engineering technical note (CHETN) is to specify the software requirements, architecture, and detailed design for the DataSwitch Data Sweeper (DS)² application. This document is designed for the software developers maintaining (DS)² and is intended to aid these developers in understanding its architecture and underlying functionality.

Background. Lock Operations Management Application (LOMA) is a system designed to collect, store, and display operational data to assist lock operations. The primary data component is the Automatic Identification System (AIS) that provides real-time and historical display of commercial river traffic. The DataSwitch applications provide connectivity between the LOMA servers and the transceivers deployed in the field. This document is intended for the personnel involved in the development and maintenance of both LOMA and (DS)².

Scope. The boundary of this document is the custom software application known as (DS)². While the service, logs, and remote database are important components of the overall system, they are not considered to be components of (DS)². They are components residing outside of (DS)² with which it must interface.

Overview. (DS)² will collect the daily DataSwitch logs, remove any malformed data, then relationally store the data for historical record and analysis.

SOFTWARE-SPECIFIC REQUIREMENTS SPECIFICATION: Figure 1 shows the use-case diagram for the application. The service initiates the application execution. The database and file system are passive actors. The application retrieves and adds logs to the file system. It also pushes data into the database.

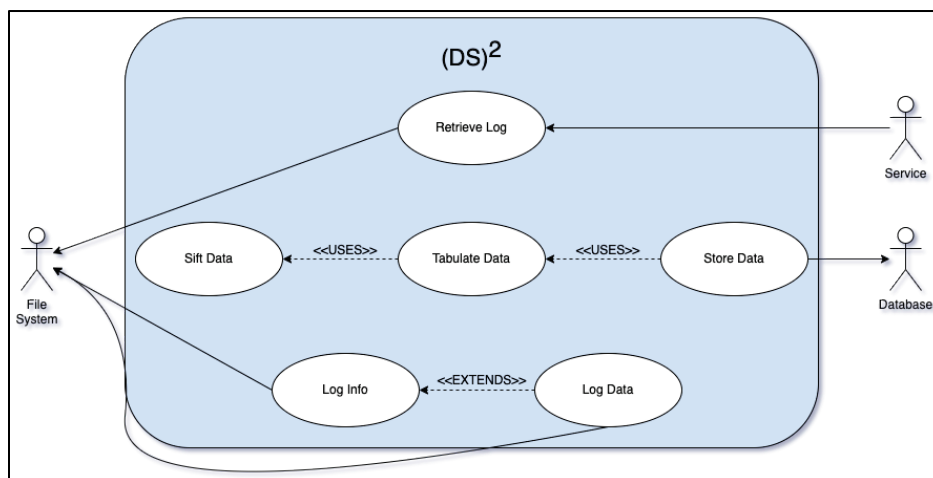


Figure 1. Use-case diagram.

Details pertaining to the software specific requirements specification may be found in IEEE (1998).

USE CASES: Table 1 through Table 6 detail the use cases for this application.

Table 1. Retrieve log.	
Description	Retrieves a log for processing
Trigger	Service
Flow	The application checks if any logs exist. If any exist, it opens one for reading.
Alternate Flow(s)	If no logs exist, the application logs the information and ends execution.

Table 2. Sift data.	
Description	Filters unwanted data from each log
Trigger	Application flow
Flow	The application reads the log line by line until it locates a desired line.
Alternate Flow(s)	The application has read to the end of the log, so it closes and deletes the log.
Precondition(s)	A log exists that needs to be processed.

Table 3. Tabulate data.	
Description	Tabulates desired data
Trigger	Application flow
Flow	The application splits the data, verified each datum, then tabulates the data.
Alternate Flow(s)	The application splits the data then detects a malformed datum so discontinues check.
Precondition(s)	A line of data exists to be processed.

Table 4. Store data.	
Description	Stores the tabulated data in the database
Trigger	Application flow
Flow	The application stores the row in the database.
Alternate Flow(s)	The application cannot connect to the database so aborts attempting to store the row. The application fails to store the row so aborts attempting to do so.
Precondition(s)	A row of data exists to be stored.

Table 5. Log data.	
Description	Logs the row which failed to store
Trigger	Database failure
Flow	The application logs the row to a separate, application-created log.
Precondition(s)	Row exists that failed to store.

Table 6. Log information.	
Description	Logs the state of the application execution
Trigger	Application flow
Flow	The application logs pertinent information.
Precondition(s)	Situation occurred befitting a note in the application log.

SOFTWARE ARCHITECTURAL DESIGN

Architecture. (DS)² resides on all LOMA servers where the DataSwitch is installed. (DS)² retrieves the DataSwitch Logs from the local server for analysis. After processing the logs, the application connects to the database on a remote server via Transmission Control Protocol to push the data for storage and analysis. Individual components of the system are described in Table 7.

Table 7. System components.	
Name	Function
Application Server	The server on which the application resides
Application Log	The log containing information about the execution of the application
Database	The relationally structured set of DataSwitch data
Database Server	The server on which the database resides
DataSwitch Log	A log from the DataSwitch containing the desired data
(DS) ²	The application of which this document is the focus
OS	The operating system installed on the server
Processed Log	A DataSwitch Log that has been cleaned of all unwanted data but has yet to be stored in the database

Deployment Diagram. Figure 2 shows the deployment diagram for the system. For simplicity, (DS)² resides on the server containing the DataSwitch logs needing to be processed. This gives the application ease of access to those logs. This server also interacts with the database server, so all the necessary rules and permissions are already in place.

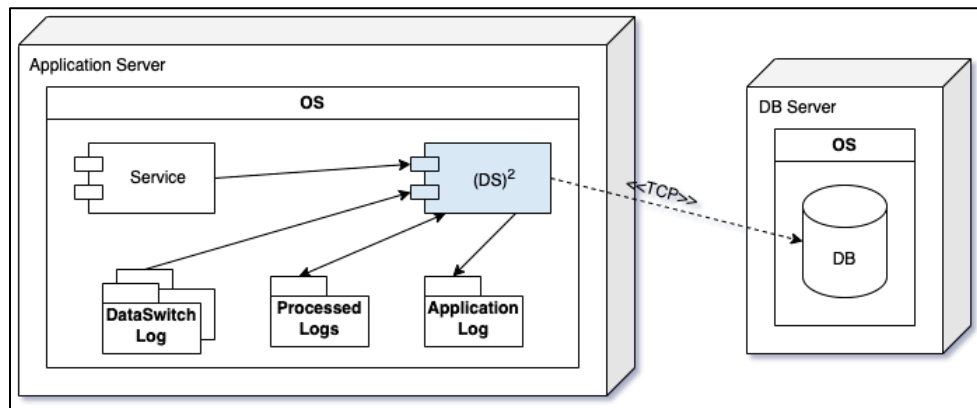


Figure 2. Deployment diagram.

DETAILED SOFTWARE DESIGN

Class Diagram. Figure 3 shows the class diagram for the system. The main program uses two helper classes: FileHelper and Logging. FileHelper provides all the logic needed to open and edit the DataSwitch logs, and Logging contains all necessary functionality to log the events of (DS)².

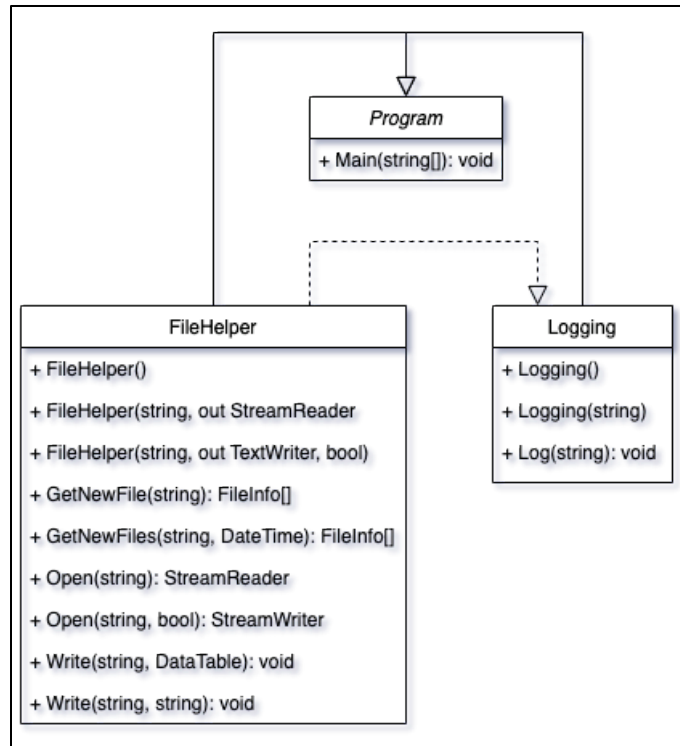


Figure 3. Class diagram.

Database Diagram. Table 8 details the database diagram of the system. The database consists of a single table with the following columns. The type of each field is identified as well as its nullability.

Table 8. Database diagram.		
NAME	VARCHAR(MAX)	NULL
VESSEL_ID	VARCHAR(10)	NOT NULL
RECEIVER_ID	VARCHAR(10)	NOT NULL
REPORT_DATE	DATE	NOT NULL
REPORT_TIME	TIME(0)	NOT NULL
REPORT_CHANNEL	CHAR(1)	NULL
RANGE	FLOAT	NULL
BEARING	FLOAT	NULL
COG	FLOAT	NULL
SOG	FLOAT	NULL
LATITUDE	FLOAT	NOT NULL
LONGITUDE	FLOAT	NOT NULL
VDM	VARCHAR(MAX)	NULL
GEOLOCATION	GEOGRAPHY	NULL

LOGIC FLOW: Flowcharts for the (DS)² functions are provided in the following pages (Figure 4 through Figure 10). These flowcharts depict the flow of logic throughout the system.

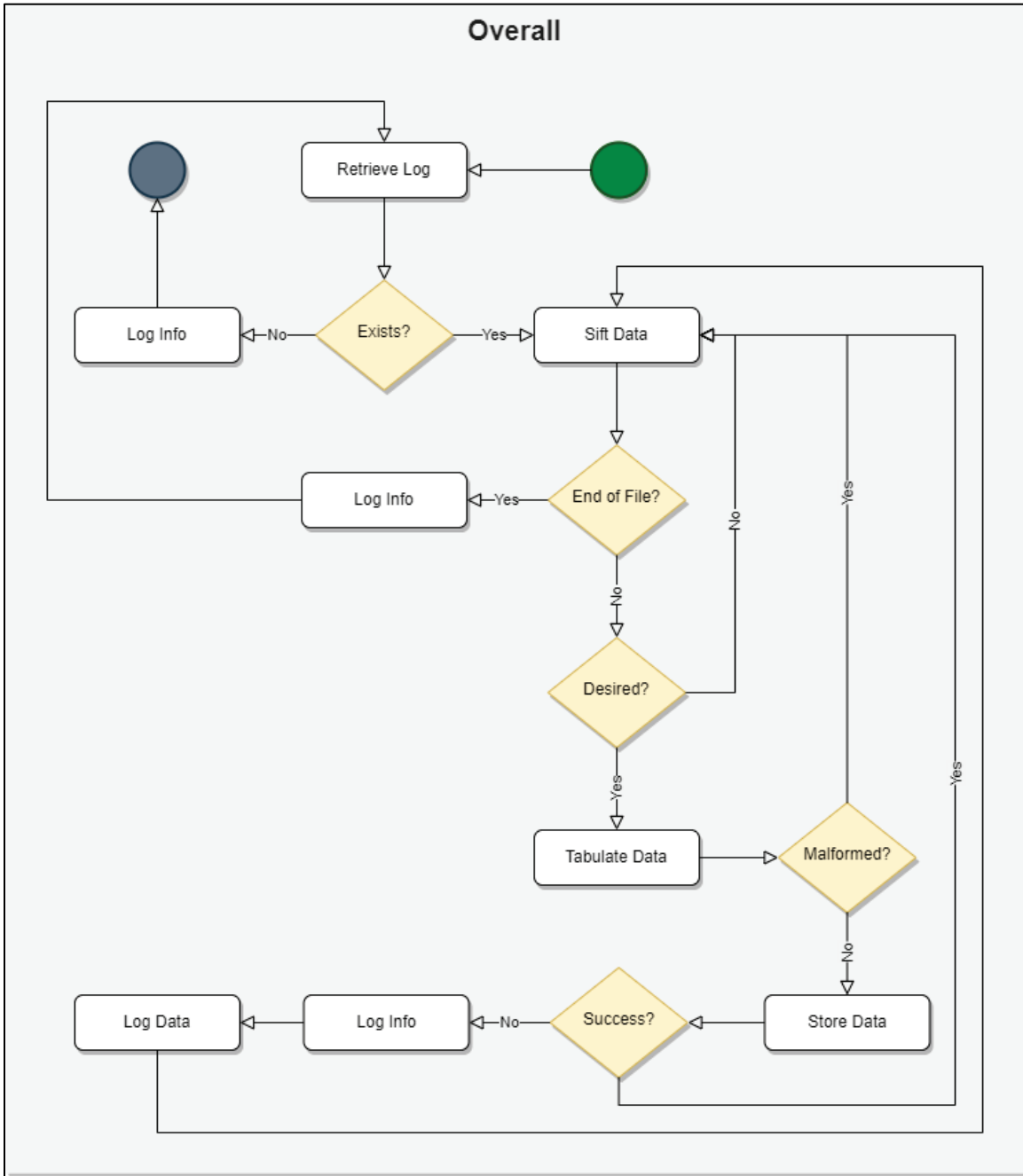


Figure 4. System flowchart.

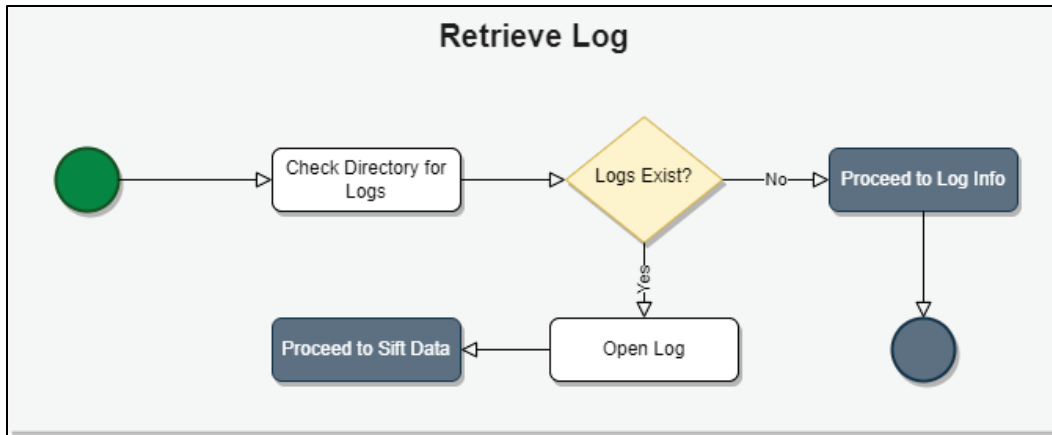


Figure 5. Log-retrieval flowchart.

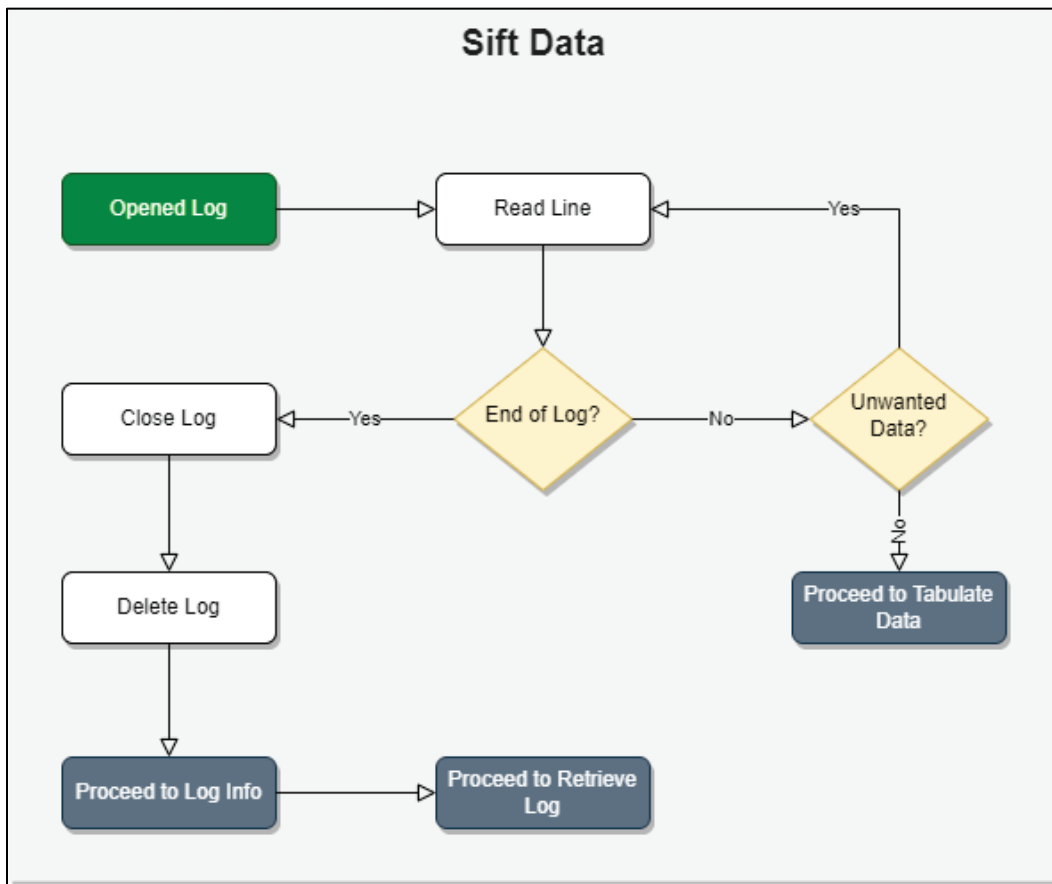


Figure 6. Data-sifting flowchart.

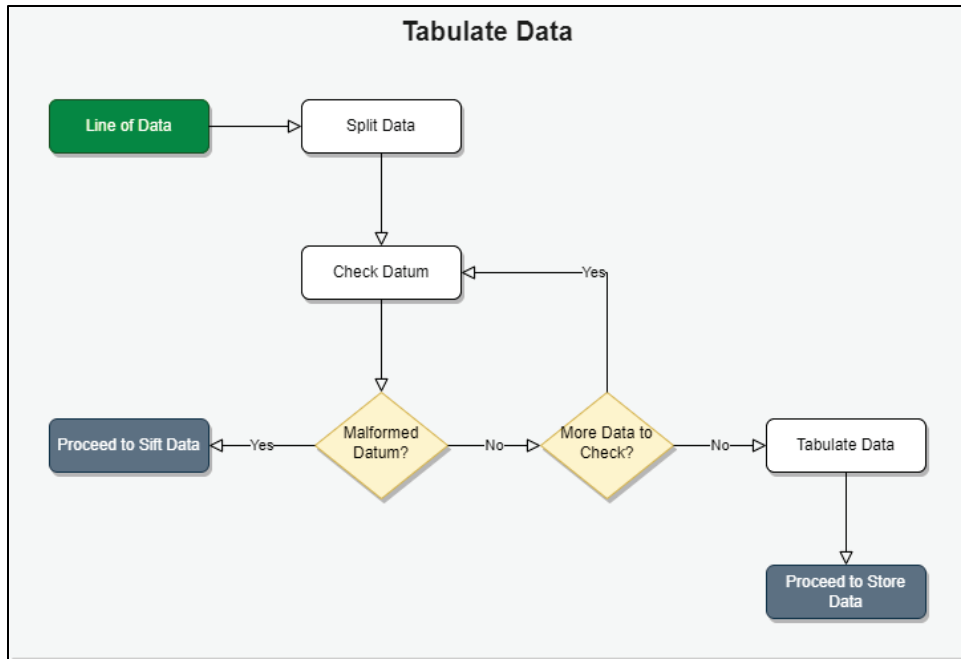


Figure 7. Data-tabulation flowchart.

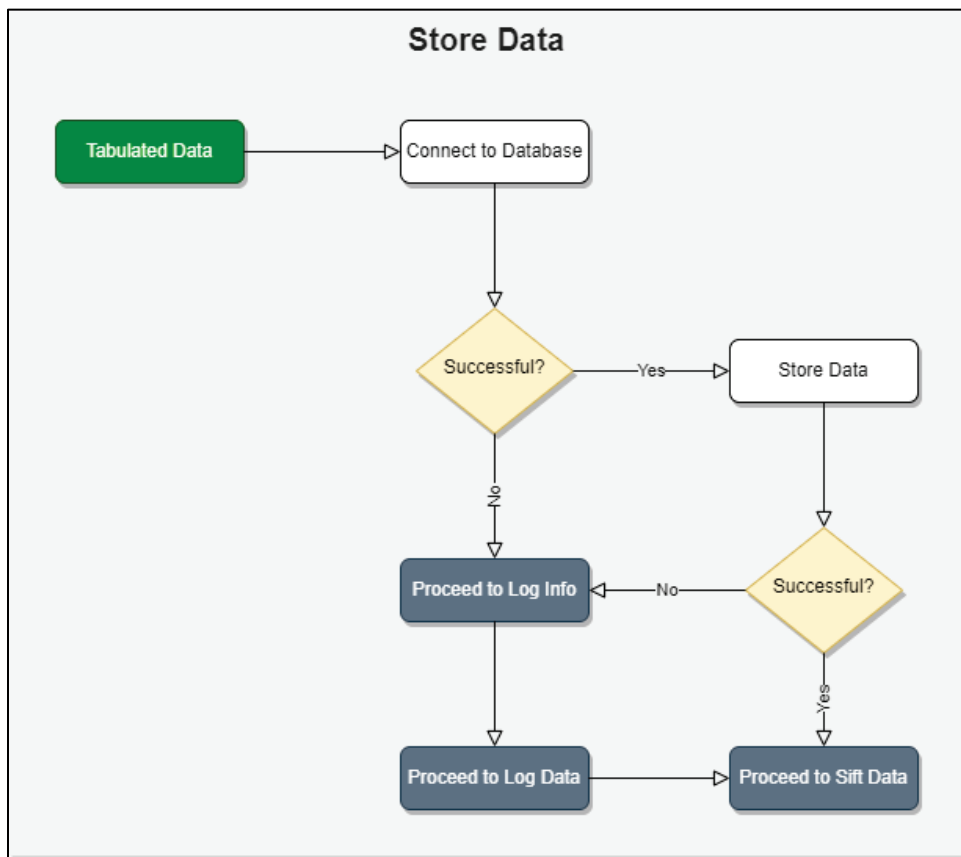


Figure 8. Data-storage flowchart.

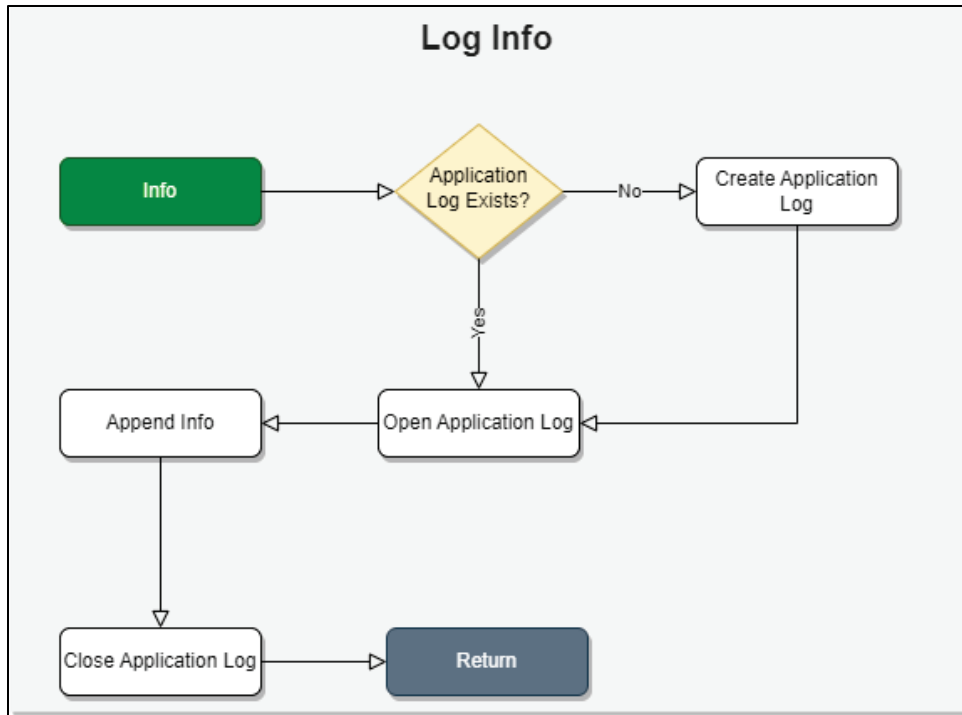


Figure 9. Information-logging flowchart.

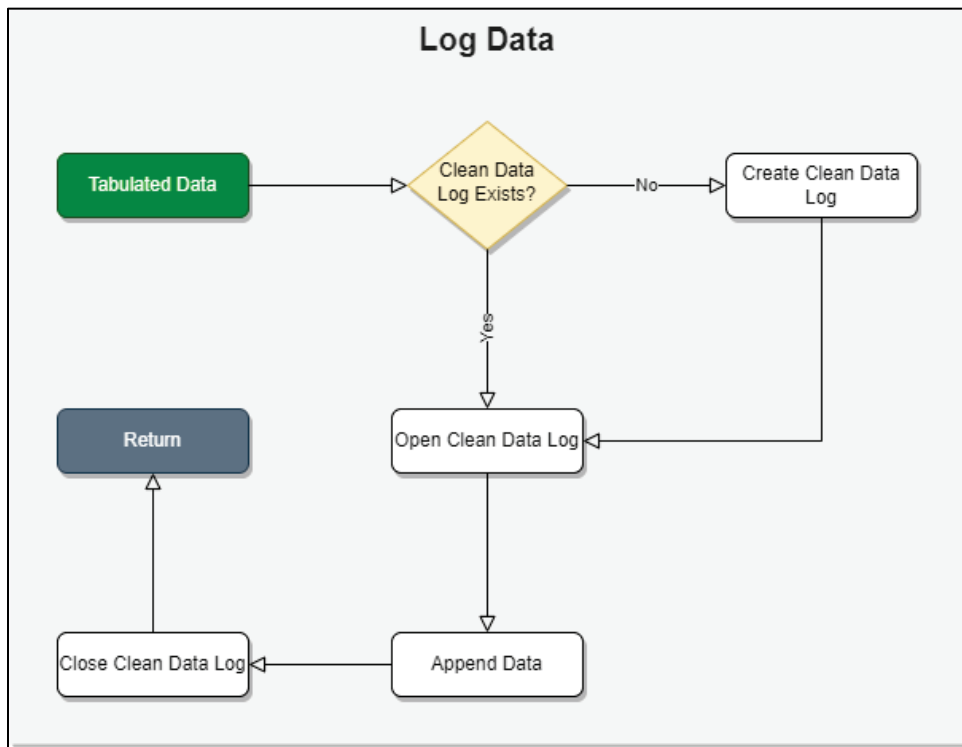


Figure 10. Data-logging flowchart.

SUMMARY: (DS)² is a system developed to assist LOMA administrators in gathering, scrubbing, and storing operational data from logs for historical record and analytics.

ADDITIONAL INFORMATION: This CHETN was prepared as part of the US Army Corps of Engineers, US Army Engineer Research and Development Center (ERDC), Navigation Systems Research Program (NavSys). The (DS)² was developed while the NavSys program manager was Mr. Charles E. Wiggins, ERDC CHL. The authors acknowledge and appreciate the input and assistance provided by other LOMA team members during (DS)² development. At the time of publication of this report, Ms. Morgan Johnston serves as the NavSys program manager, and Dr. Marin Kress serves as the LOMA program manager. Questions regarding this CHETN, LOMA, or (DS)² may be directed to the primary author and principal investigator Mr. Joseph H. Farrar, research computer scientist, ERDC Information Technology Laboratory, Joseph.H.Farrar@usace.army.mil; or to other team members Dr. Marin Kress, LOMA program manager, Marin.M.Kress@usace.army.mil; Mr. James T. Kilroy, navigation AIS specialist (CHL), James.Kilroy@usace.army.mil; Mr. Kenneth W. Swan, computer engineer (CHL), Kenneth.W.Swan@usace.army.mil; and Mr. Brady A. Towne, civil engineering technician (CHL), Brady.A.Towne@usace.army.mil.

This CHETN should be referenced as follows:

Farrar, Joseph H., and James (Lee) Whitlow. 2023. *DataSwitch Data Sweeper (DS)²*. ERDC/CHL CHETN-IX-67. Vicksburg, MS: US Army Engineer Research and Development Center. <http://dx.doi.org/10.21079/11681/47765>.

REFERENCES

IEEE (Institute for Electrical and Electronics Engineers). 1998. *Standard 830-1998; IEEE Recommended Practice for Software Requirements Specifications*. Piscataway, NJ: Institute for Electrical and Electronics Engineers.

NOTE: *The contents of this technical note are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such products.*