**California State University Long Beach**

**College of Engineering**

**Computer Engineering Computer Science Department**

**Tuesday, November 11, 2019**

**Sixfold Flat File ArchiveDesign Document V1.0**

**Group Members:**

**Michell Kuang (Team Lead) 013421094**

**Joshua McDaniel 014542786**

**Jingyan Du 014436615**

**Peter Park 002948398**

**Jacen Tan 012393782**

**Daniel Gione 016513144**

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Comments** |
| V1.0 | 11/10/2019 | Jingyan Du | -Original document |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[**Table of Contents**](#_wxmdysu4hjxy) **2**

[**1.Overview**](#_a0i7yj6g39sw) **3**

[**2. Feature Life Cycle**](#_dppy5z1zxujh) **3**

[**3. Function Called**](#_wlw8e87qtimi) **4**

[3.1.Diagram and Description](#_ajch7d45ow80) 4

[3.2.Error Handling](#_70uvgm2myciw) 4

[**4. Data Collection**](#_7fkd98yxqih4) **4**

[4.1. Diagram and Description](#_t3ad5992xym8) 4

[4.2. Error Handling](#_ky5w3p1z9bze) 4

[**5. Flat File Output**](#_bo37lj2ij9so) **4**

[5.1 Diagram and Description](#_mcjvgnuhw4hg) 4

[5.2 Error Handling](#_c8ukex7f36o5) 4

[**6. File Store**](#_5hosekz1woq4) **4**

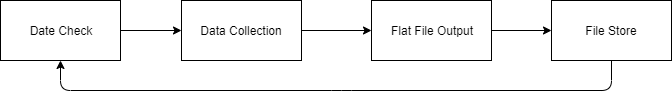
[6.1 Diagram and Description](#_b26i6sj5p3d8) 4

[6.2 Error Handling](#_912kmim9398w) 4

# Overview

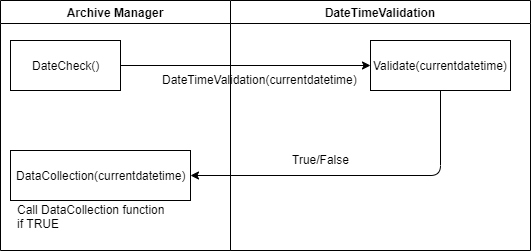
This Document is made to show design conecpts of flat file data archive feature of SixFold Household Manager. To use storage space efficiently, archive systems for logs are necessary. Diagrams for data flowing and error handling will be included to guide the developers.

# Feature Life Cycle



# Date Check

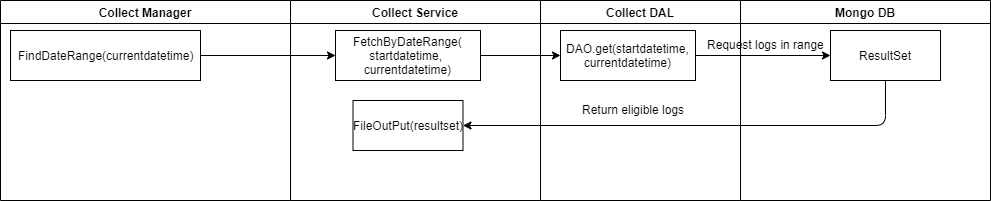
## Diagram and Description



The flat file archive shall start on the 1st of each month at 00:00:00:00 UTC, system shall always keep checking if the datetime is eligible to do a flat file data archive. When DateCheck() function is called, it will request a validation to check if the datetime is correct and then return the result. If a TRUE is received, FindDateRange() function shall be called with the current datetime as a parameter to process the data collection phase. If a False is received, no function should be called and the system shall start another checking period.

# Data Collection

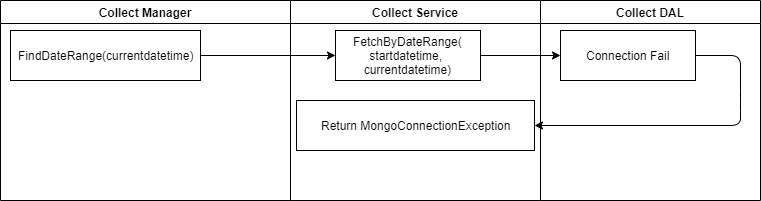
## Diagram and Description



Once the date is validated, the FindDateRange() function shall be called to find a range for logs those are eligible to be archived. Then the system shall request the result set from database based on the datetime range. The result set shall be return as an array.

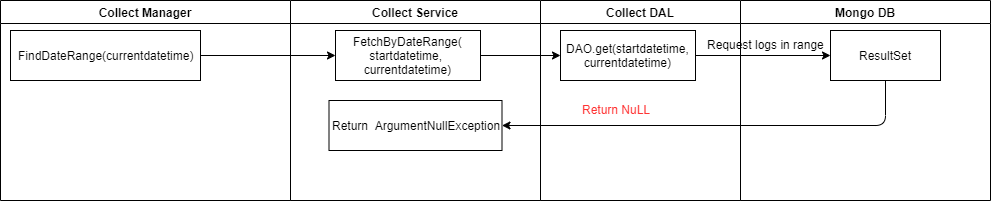
## Error Handling

### Mongo DB Connection Exception



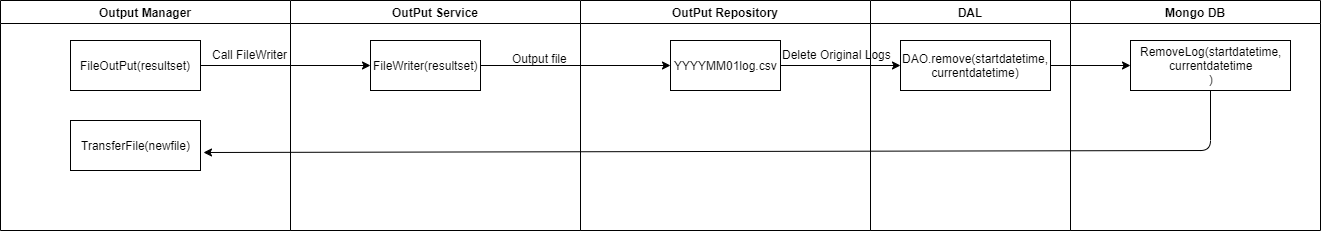
If the connection to Mongo DB is failed, the FetchByDateRange() function should throw a MongoConnectionException. After log the error, system should retry connecting to Mongo DB for max of 3 times. If the third time still fail, system should stop this archive, log archive as failure and notify the admin.

### Argument Null Exception



If the result set from Mogo DB is null, it means no logs could be found based on the datetime range. In this way, the FileOuPut() function should not create a flat file for the empty array. The FileOutPut() function shall throw an argumentNullException, and the archive process should be stopped since no files could be created. The system admin shall be notified that no logs could be archived.

# Flat File Output

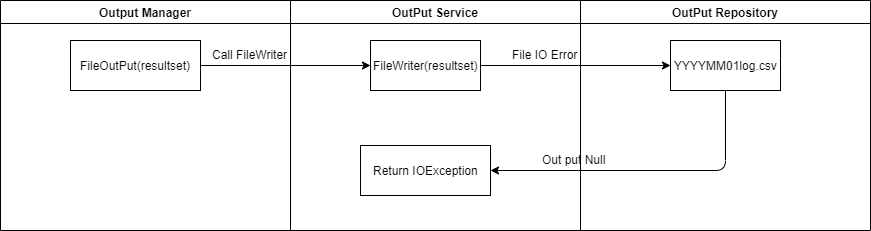


## Diagram and Description

Once the result set is returned, system shall write the result set to a .csv file for storage. If file is written successfully, the original logs shall be removed to save storage space.

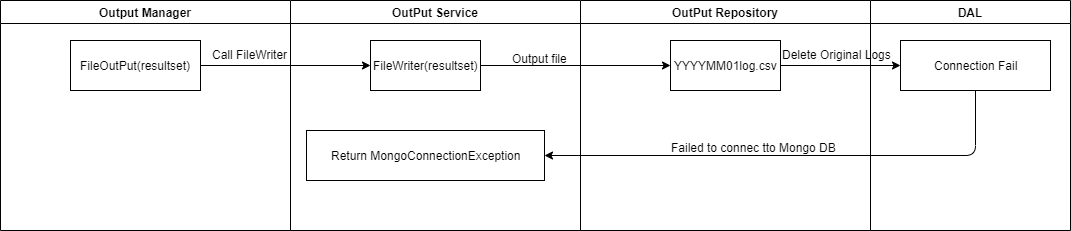
## Error Handling

File IOException



If the file output is failed, an IOException shall be throw and original data should not be removed. The system shall retry to create .csv file after logging the error. Up to 3 times if the file output is still failed. The process of data archive should be stopped and the system admin shall be notified.

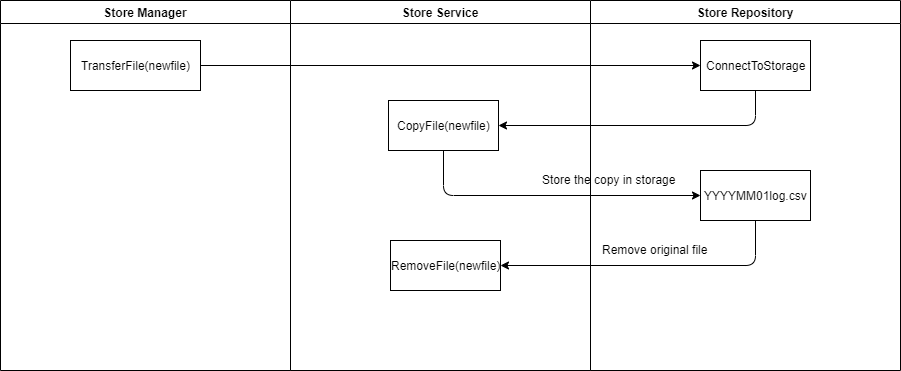
### Mongo DB Connection Exception



If the connection to Mongo DB is failed, then no data could be used to create a flat file. The function shall throw a MongoConnectionException, the error shall be logged and no original logs should be removed. The system shall retry the connection up to 3 times. If the connection is still failed, this archive process shall be stopped and logged as failure, the system admin shall be notified.

# File Store

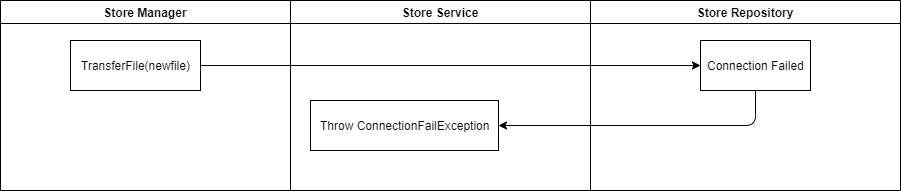
## Diagram and Description



System shall try to connect to the storage machine and make a copy of the original flat file. Once the flat file is copied successfully, the original flat file shall be deleted. The period of data archive is finished and it shall be logged as success.

## Error Handling

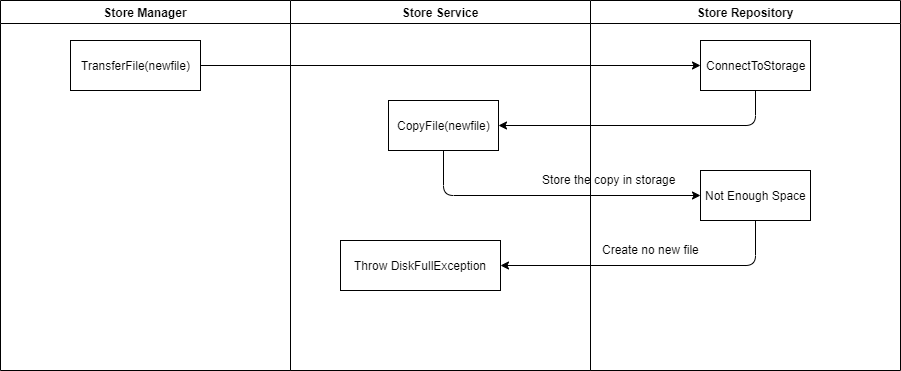
### Connection Fail Exception



If the storage machine does not response, it shall be seen as a connection fail, a ConnectionFailException shall be thrown. The system shall log the error and try to reconnect to to the storage machine and if it still failed up to 3 times, the archive process shall be

stopped. The system shall notify the system admin and log the data archive as failure.

### Disk Full Exception



If the disk of storage machine is full, no flat file copy could be created, the function shall throw a DisckFullException and notify the system admin. The data archive process shall be stopped to avoid data lost. And this process of data archive shall be logged as failure.