# **Use Case** UC-005

Version 1.0

## **Revision History**

| Date | Author | Description of change |
| --- | --- | --- |
| 10/19/2023 | Jaeden S. | Use case created |
| 10/19/2023 | Devon V. | Minor Edits and tidying |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Use Case:** Archiving obsolete data

**Id**: UC-005

**Description**

The Data Management team receives a notification of data that is obsolete and is going to be removed from the database. The Database Administrator will then archive this data for potential later use.

**Level:** User Goal

**Primary Actor**

Database Administrator

**Supporting Actors**

Data Analyst

Data Management Team

Database Administrator

**Stakeholders and Interests**

Data Analyst: is interested in storing the data that is potentially going to be removed, as the data could be beneficial in future research.

Data Management Team: is interested in maintaining an up-to-date dataset, any obsolete or stale data is removed to ensure the Data Analysis Team has only valid data to work with.

Data Administrator: is interested in maintaining the system’s stability, consistency, and security. Sensitive data being archived will need to be stored securely so that it is not accessible publicly. The System Administrator is also interested in ensuring the archive process aligns with the technical requirements.

**Pre-Conditions**

Database platform is online.

The Data Management Team has access to the database via phpMyAdmin.

The data being removed is valid to archive.

There is a secure place to archive the data.

**Post Conditions**

Success end condition

The Data Management Team is notified of the obsolete dataset that will be removed and they make an archive request to the Database Administrator. The dataset is then archived successfully in a secure location by the Database Administrator for potential future use.

Failure end condition:

The Data Administrator is unable to securely archive the dataset, the obsolete data is removed with no archive to access for future use.

Minimal Guarantee

The minimal guarantee is the data management team will be notified that data is going to be removed within a week.

**Trigger**

The data management team is notified that some data will be removed.

## **Main Success Scenario**

1. The Data Management Team is notified that an obsolete dataset is going to be removed in a week.
2. The Data Management Team informs the Data Administrator of the dataset to be removed.
3. The Data Administrator locates a secure location to store the data.
4. The Data Administrator archives the dataset that is to be removed in the secure location.
5. The dataset is safely archived for potential future use.

## **Extensions**

1a. The Data Management Team neglects/forgets to notify higher authority

1. The dataset is removed from the database and is lost indefinitely.
2. This is considered a failure end condition.

3a. The Data Administrator is unable to find a secure location to archive the dataset

1. An insecure location is located instead.
2. Proceed from step 4, using an insecure location in place of a secure one.

3b. The Data Administrator is unable to find any location to archive the dataset

1. The dataset is removed from the database and is lost indefinitely.
2. This is considered a failure end condition.

4a. Saving the data is unsuccessful

1. The data is removed from the database
2. There is no data archived
3. If the data is needed for later use it is no longer accessible
4. This is considered a failure end condition.

**Frequency:** Once per month or when obsolete data is to be removed

**Assumptions**

The system will alert its users of deletions one week prior

The Data Administrator and Data management team has access to phpMyAdmin

The users can extract a snapshot of the database to archive

The obsolete data may potentially be used in the future

## **Special Requirements**

**Performance:**

1.Extracting from the database should be processed efficiently

2.Users connecting to phpMyAdmin should have valid login credentials

**User Interface:**

1.User should know how to access UI

2.User should know how to extract data from UI

**Security:**

1.Any data extracted and archived should be stored in a secure location

**Logging:**

1. An associated log must be created with the time and location that data was archived

## **Issues**

1. What is deemed a valid dataset to archive?
2. What are the security requirements for archiving data?
3. Does the system have storage limitations for archiving?

## **To do**

1. Find when data is to be removed from the database.
2. Determine when the notification for removal is triggered.
3. Resolve whether or not the process is automatic or can be made automatic, and if it is, add an alternative workflow
4. Resolve how long will be allotted from when the notification is given before data is permanently lost.