**Archival Databases Use Case Draft - 3**

**Team Members:**

Durga Sparsha Bhagwat

Pavan Jupally

Shafali Gondi

Raihan Ahmed Mohammed

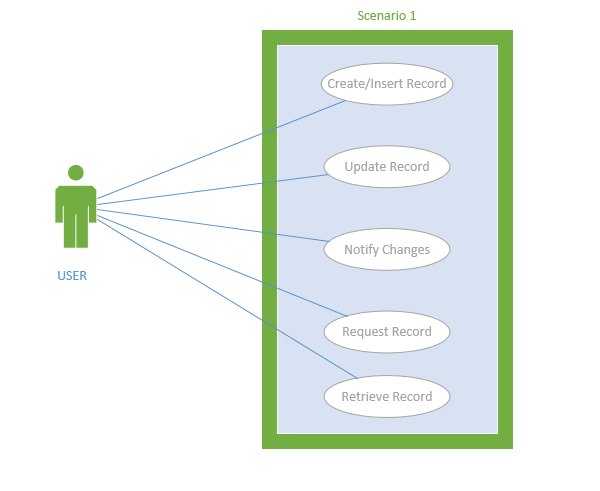
**Use Case Draft – 3:**

This use case diagram(s) depicts the various interactions of the user with the database. The User is the main actor in all the scenarios since the user will be directly interacting with the UI which is directly interacting with the database. The scenarios are defined based on the exchange of data with the database and the user.

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Goal-Level** | **Description** |
| ++ | Very High Summary | Rachel doing checklist on interface |
| + | Summary | Displaying reports on what actions are performed on files and what are the action to be performed |
| ! | User Goal | Reports generated by fixity |
| - | Subfunction | The interface talks to bagger and fixity |
| -- | Too Low | Adding annotations and comments along with reports to database |

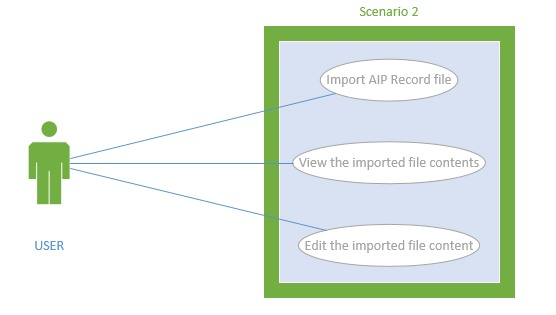
**Scenario 1:**

The user enters the AIP record into the system which contains information like whether the files are bagged, uploaded on Duracloud, AIPed, Fixity checks performed, etc. This information is saved in the database as tuples in single or multiple tables. User can request existing record from the database, the system retrieves the records and displays it in the UI. The user can update records, if there are any changes to be made and these changes are saved in database. The user is notified about the record being updated/created in the database.



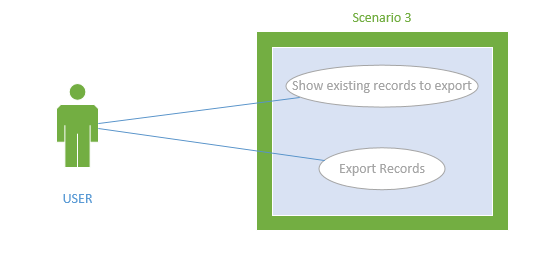
**Scenario 2:**

The user has AIP records which is supposed to be inserted into the database. The system reads the file and adds it to the database. After the user has imported the record the user can view/edit the imported file contents in the UI which are populated from the database.



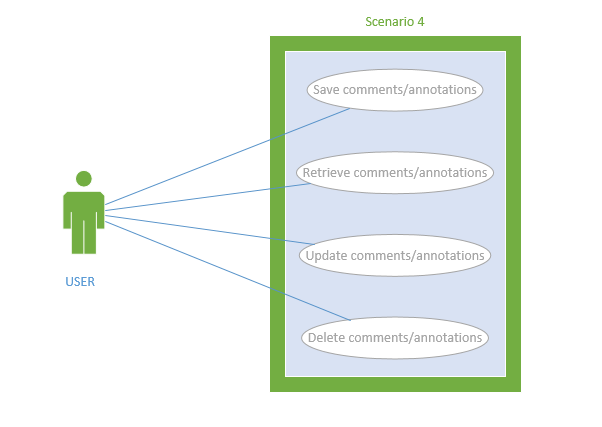
**Scenario 3:**

The user can browse existing records and export an AIP record file from the database.



**Scenario 4:**

The user adds comments/annotations to the UI which is in turn saved in database, this can be retrieved, updated and deleted based on requirement.

****