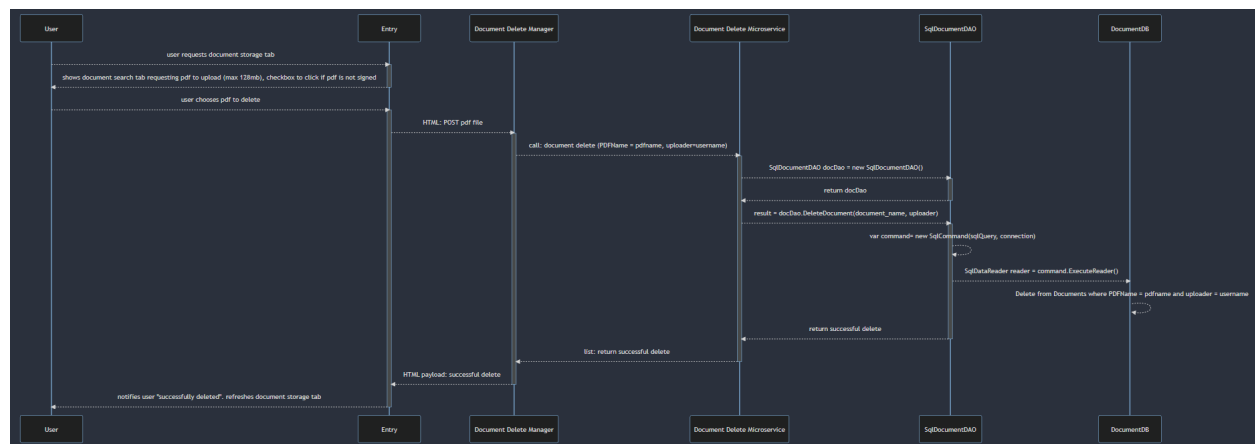


success



sequenceDiagram

User-->>+Entry: user requests document storage tab

Entry-->>-User: shows document search tab requesting pdf to upload (max 128mb), checkbox to click if pdf is not signed

User-->>+Entry: user chooses pdf to delete

Entry-->>+Document Delete Manager: HTML: POST pdf file

Document Delete Manager-->>+Document Delete Microservice: call: document delete (PDFName = pdfname, uploader=username)

Document Delete Microservice-->>+SqlDocumentDAO: SqlDocumentDAO docDao = new SqlDocumentDAO();

SqlDocumentDAO-->>-Document Delete Microservice: return docDao

Document Delete Microservice-->>+SqlDocumentDAO: result = docDao.DeleteDocument(document\_name, uploader)

SqlDocumentDAO-->>SqlDocumentDAO: var command= new SqlCommand(sqlQuery, connection)

SqlDocumentDAO-->>+DocumentDB: SqlDataReader reader = command.ExecuteReader()

DocumentDB-->>DocumentDB: Delete from Documents where PDFName = pdfname and uploader = username

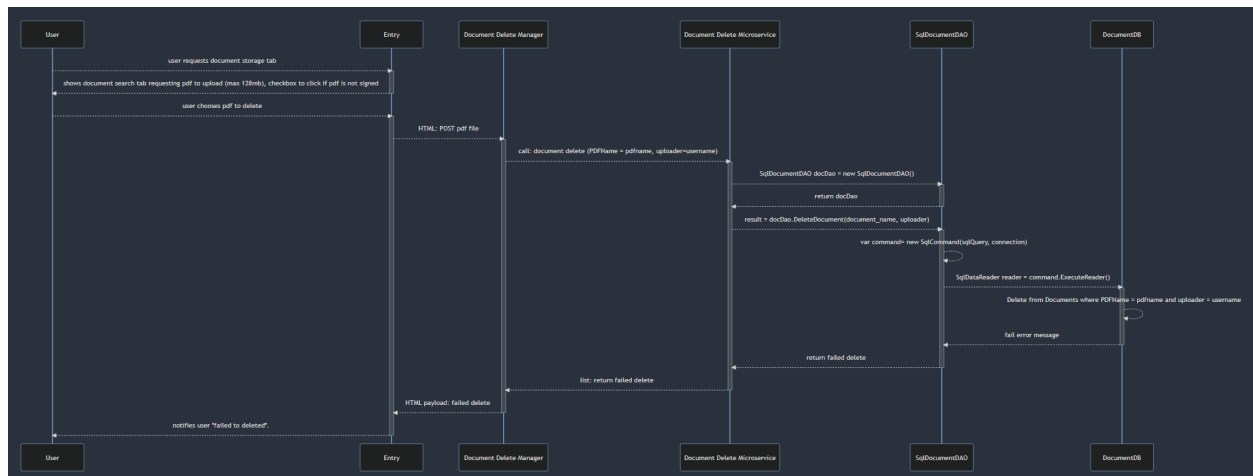
SqlDocumentDAO-->>-Document Delete Microservice: return successful delete

Document Delete Microservice-->>-Document Delete Manager: list: return successful delete

Document Delete Manager-->>-Entry: HTML payload: successful delete

Entry-->>-User: notifies user "successfully deleted". refreshes document storage tab

## Fail 1



### sequenceDiagram

User-->>+Entry: user requests document storage tab

Entry-->>-User: shows document search tab requesting pdf to upload (max 128mb), checkbox to click if pdf is not signed

User-->>+Entry: user chooses pdf to delete

Entry-->>+Document Delete Manager: HTML: POST pdf file

Document Delete Manager-->>+Document Delete Microservice: call: document delete (PDFName = pdfname, uploader=username)

Document Delete Microservice-->>+SqlDocumentDAO: SqlDocumentDAO docDao = new SqlDocumentDAO();

SqlDocumentDAO-->>-Document Delete Microservice: return docDao

Document Delete Microservice-->>+SqlDocumentDAO: result = docDao.DeleteDocument(document\_name, uploader)

SqlDocumentDAO-->>SqlDocumentDAO: var command= new SqlCommand(sqlQuery, connection)

SqlDocumentDAO-->>+DocumentDB: SqlDataReader reader = command.ExecuteReader()

DocumentDB-->>DocumentDB: Delete from Documents where PDFName = pdfname and uploader = username

DocumentDB-->>-SqlDocumentDAO: fail error message

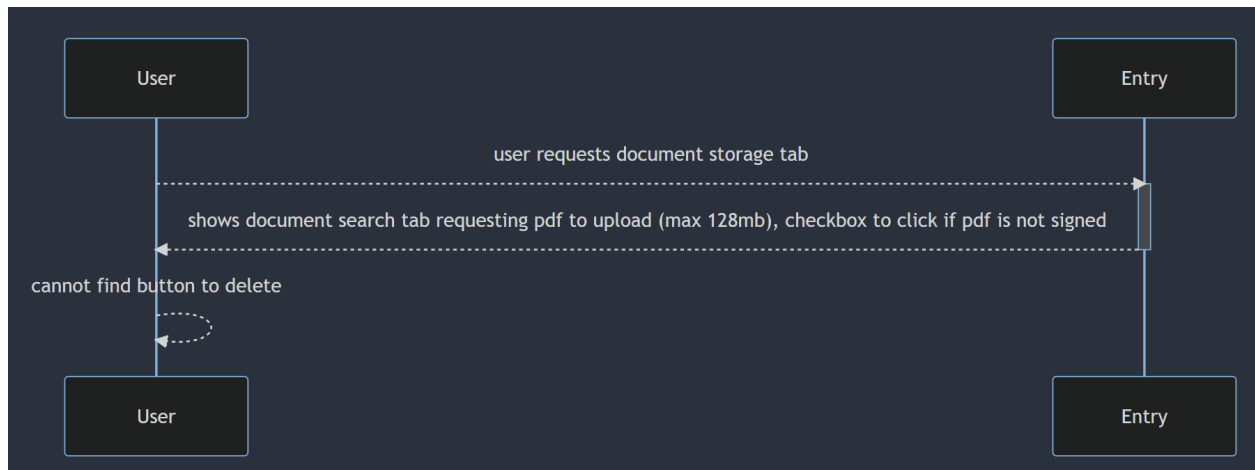
SqlDocumentDAO-->>-Document Delete Microservice: return failed delete

Document Delete Microservice-->>-Document Delete Manager: list: return failed delete

Document Delete Manager-->>-Entry: HTML payload: failed delete

Entry-->>-User: notifies user "failed to delete".

## Fail 2



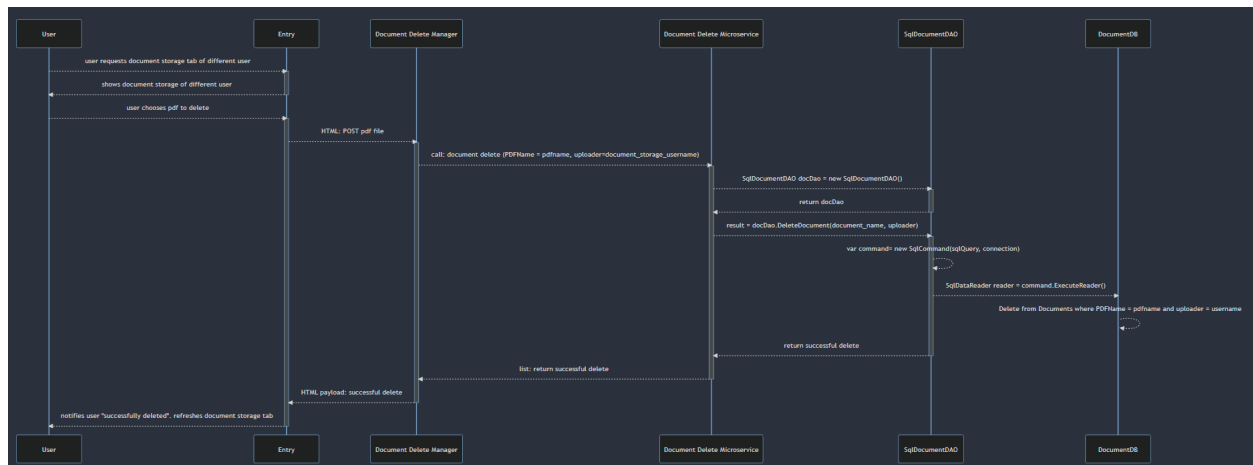
sequenceDiagram

User-->>+Entry: user requests document storage tab

Entry-->>-User: shows document search tab requesting pdf to upload (max 128mb),  
checkbox to click if pdf is not signed

User-->>User: cannot find button to delete specific pdf

## Fail 3



## sequenceDiagram

User-->>+Entry: user requests document storage tab of different user

Entry-->>-User: shows document storage of different user

User-->>+Entry: user chooses pdf to delete

Entry-->>+Document Delete Manager: HTML: POST pdf file

Document Delete Manager-->>+Document Delete Microservice: call: document delete (PDFName = pdfname, uploader=document\_storage\_username)

Document Delete Microservice-->>+SqlDocumentDAO: SqlDocumentDAO docDao = new SqlDocumentDAO();

SqlDocumentDAO-->>-Document Delete Microservice: return docDao

Document Delete Microservice-->>+SqlDocumentDAO: result = docDao.DeleteDocument(document\_name, uploader)

SqlDocumentDAO-->>SqlDocumentDAO: var command= new SqlCommand(sqlQuery, connection)

SqlDocumentDAO-->>+DocumentDB: SqlDataReader reader = command.ExecuteReader()

DocumentDB-->>DocumentDB: Delete from Documents where PDFName = pdfname and uploader = username

SqlDocumentDAO-->>-Document Delete Microservice: return successful delete

Document Delete Microservice-->>-Document Delete Manager: list: return successful delete

Document Delete Manager-->>-Entry: HTML payload: successful delete

Entry-->>-User: notifies user "successfully deleted". refreshes document storage tab