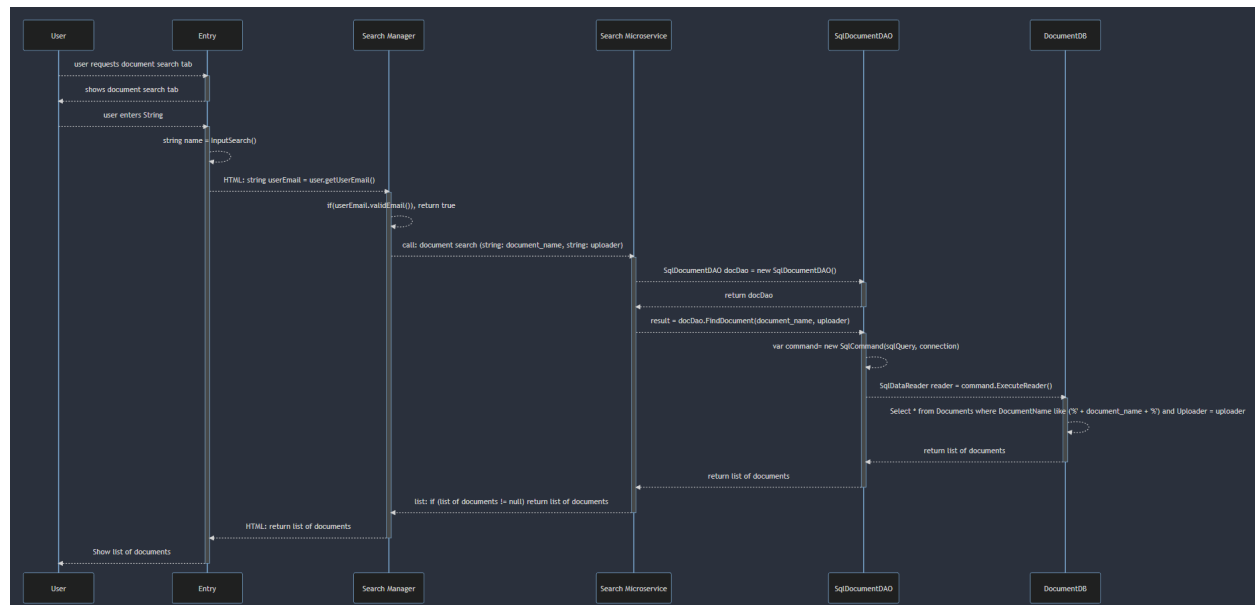


Success 1



sequenceDiagram

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

Entry-->>-User: shows document search tab

User-->>+Entry: user enters String

Entry-->>Entry: string name = InputSearch()

Entry-->>+Search Manager: HTML: string userEmail = user.getUserEmail()

Search Manager-->>Search Manager: if(userEmail.validEmail()), return true

Search Manager-->>+Search Microservice: call: document search (string: document_name, string: uploader)

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

SqlDocumentDAO-->>-Search Microservice: return docDao

Search Microservice-->>+SqlDocumentDAO: result = docDao.FindDocument(document_name, uploader)

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

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

DocumentDB-->>DocumentDB: Select * from Documents where DocumentName like ('%' + document_name + '%') and (Uploader = uploader or Shared = uploader)

DocumentDB-->>-SqlDocumentDAO: return list of documents

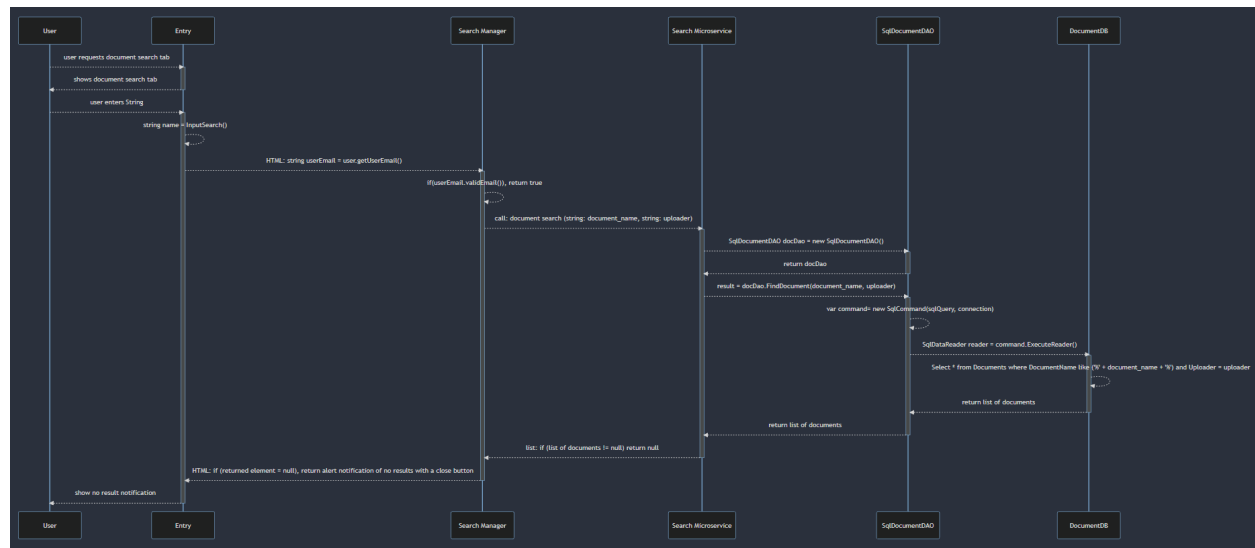
SqlDocumentDAO-->>-Search Microservice: return list of documents

Search Microservice-->>-Search Manager: list: if (list of documents != null) return list of documents

Search Manager-->>-Entry: HTML: return list of documents

Entry-->>-User: Show list of documents

Success 2



sequenceDiagram

User-->>Entry: user requests document search tab

Entry-->>User: shows document search tab

User-->>Entry: user enters String

Entry-->>Entry: string name = InputSearch()

Entry-->>Search Manager: HTML: string userEmail = user.getUserEmail()

Search Manager-->>Search Manager: if(userEmail.validEmail()), return true

Search Manager-->>Search Microservice: call: document search (string: document_name, string: uploader)

Search Microservice-->>SqlDocumentDAO: SqlDocumentDAO docDao = new SqlDocumentDAO();

SqlDocumentDAO-->>Search Microservice: return docDao

Search Microservice-->>SqlDocumentDAO: result = docDao.FindDocument(document_name, uploader)

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

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

DocumentDB-->>DocumentDB: Select * from Documents where DocumentName like ('%' + document_name + '%') and (Uploader = uploader or Shared = uploader)

DocumentDB-->>SqlDocumentDAO: return list of documents

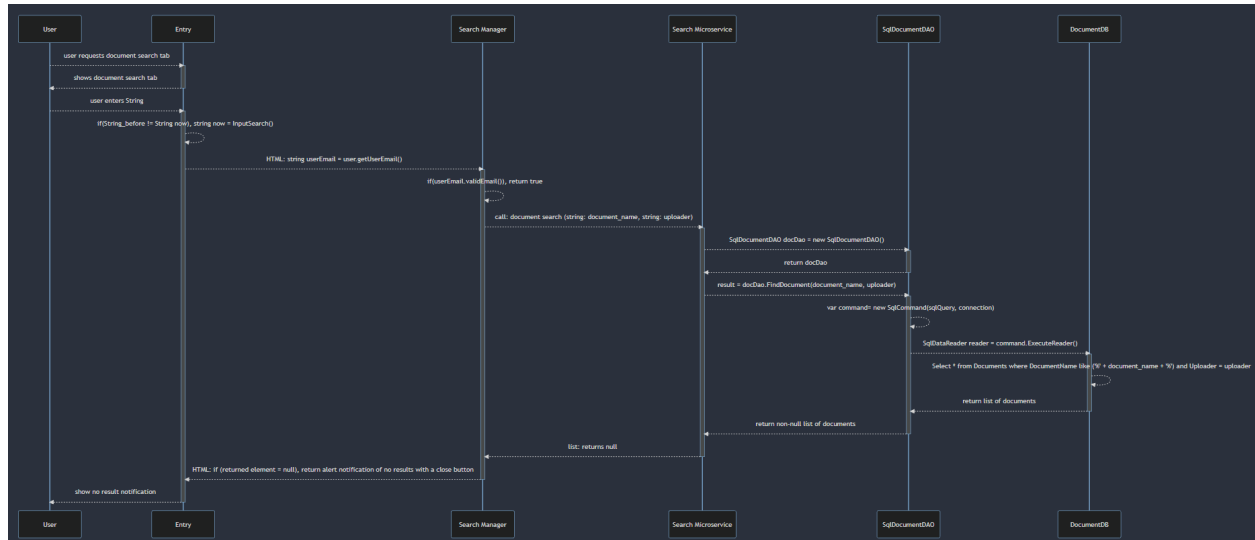
SqlDocumentDAO-->>Search Microservice: return list of documents

Search Microservice-->>Search Manager: list: if (list of documents = null) return null

Search Manager-->>Entry: HTML: if (returned element = null), return alert notification of no results with a close button

Entry-->>User: show no result notification

Fail 1



sequenceDiagram

User-->>Entry: user requests document search tab

Entry-->>User: shows document search tab

User-->>Entry: user enters String

Entry-->>Entry: if(String_before != String now), string now = InputSearch()

Entry-->>Search Manager: HTML: string userEmail = user.getUserEmail()

Search Manager-->>Search Manager: if(userEmail.validEmail()), return true

Search Manager-->>Search Microservice: call: document search (string: document_name, string: uploader)

Search Microservice-->>SqlDocumentDAO: SqlDocumentDAO docDao = new SqlDocumentDAO();

SqlDocumentDAO-->>Search Microservice: return docDao

Search Microservice-->>SqlDocumentDAO: result = docDao.FindDocument(document_name, uploader)

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

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

DocumentDB-->>DocumentDB: Select * from Documents where DocumentName like ('%' + document_name + '%') and (Uploader = uploader or Shared = uploader)

DocumentDB-->>SqlDocumentDAO: return list of documents

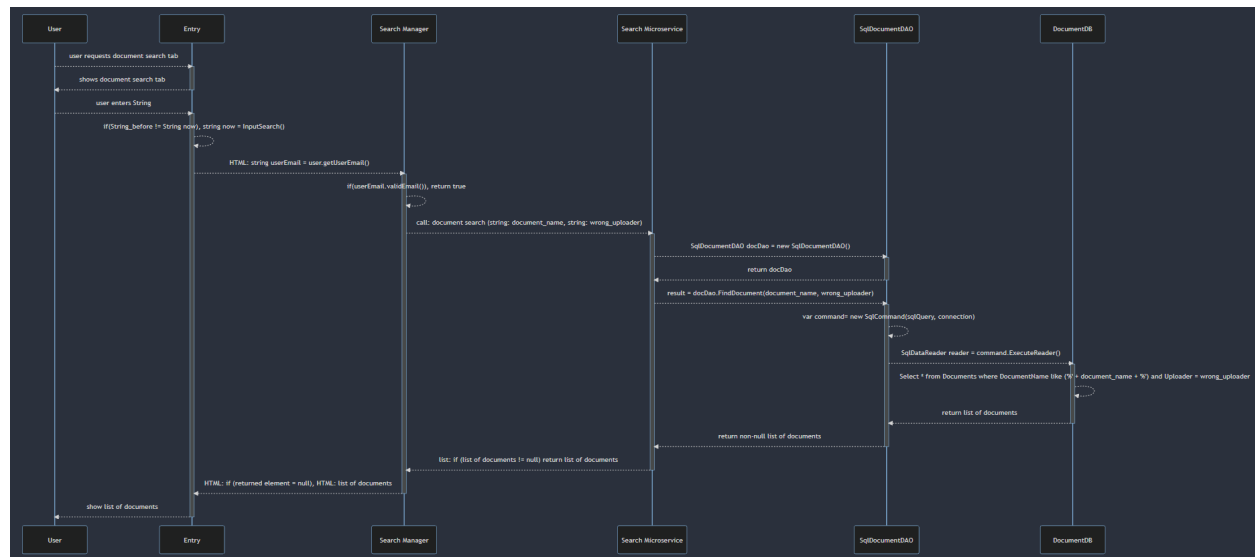
SqlDocumentDAO-->>Search Microservice: return non-null list of documents

Search Microservice-->>Search Manager: list: returns null

Search Manager-->>Entry: HTML: if (returned element = null), return alert notification of no results with a close button

Entry-->>User: show no result notification

Fail 2



sequenceDiagram

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

Entry-->>-User: shows document search tab

User-->>+Entry: user enters String

Entry-->>Entry: if(String_before != String now), string now = InputSearch()

Entry-->>+Search Manager: HTML: string userEmail = user.getUserEmail()

Search Manager-->>Search Manager: if(userEmail.validEmail()), return true

Search Manager-->>+Search Microservice: call: document search (string: document_name, string: wrong_uploader)

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

SqlDocumentDAO-->>-Search Microservice: return docDao

Search Microservice-->>+SqlDocumentDAO: result = docDao.FindDocument(document_name, wrong_uploader)

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

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

DocumentDB-->>DocumentDB: Select * from Documents where DocumentName like ('%' + document_name + '%') and (Uploader = wrong_uploader or Shared = wrong_uploader)

DocumentDB-->>-SqlDocumentDAO: return list of documents

SqlDocumentDAO-->>-Search Microservice: return non-null list of documents

Search Microservice-->>-Search Manager: list: if (list of documents != null) return list of documents

Search Manager-->>-Entry: HTML: if (returned element = null), HTML: list of documents

Entry-->>-User: show list of documents

Fail 3



sequenceDiagram

User-->>Entry: user requests document search tab

Entry-->>User: shows document search tab

User-->>Entry: user enters String

Entry-->>Entry: if(String_before != String now), string now = InputSearch()

Entry-->>Search Manager: HTML: string userEmail = user.getUserEmail()

Search Manager-->>Search Manager: if(userEmail.validEmail()), return true

Search Manager-->>Search Microservice: call: document search (string: document_name, string: wrong_uploader)

Search Microservice-->>SqlDocumentDAO: SqlDocumentDAO docDao = new SqlDocumentDAO();

SqlDocumentDAO-->>Search Microservice: return docDao

Search Microservice-->>SqlDocumentDAO: result = docDao.FindDocument(document_name, wrong_uploader)

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

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

DocumentDB-->>DocumentDB: Select * from Documents where DocumentName like ('%' + document_name + '%')

SqlDocumentDAO-->>Search Microservice: return non-null list of documents

Search Microservice-->>Search Manager: list: if (list of documents != null) return list of documents

Search Manager-->>Entry: HTML: if (returned element = null), HTML: list of documents

Entry-->>User: show list of documents