**Design document**

**1- Technologies used at the backend:**

- language : C#

- rest service: ASP.NET Web APIs

- authentication & authorization: Microsoft identity

- database: Microsoft SQL Server

- ORM: Entity framework

- DI Container: Ninject

- Architecture: N Layer

- Logger: log4net

- Unit test MS Unit test Framework

- mock framework: MOQ

**2- Design patterns used at the backend:**

* Depend on the abstraction when save the document by implement Interface class IDocumentManager and add new implementation for method[ void UploadFiles(Document doc, Stream stream )]

Then inject the new implementation using Nninject Container

-Repository:

to separate the business Layer from the Data access layer and the used ORM and make no change at the Business if we change the ORM at the future.

 -Unit of work:

to apply transactions over multiple operations and change the ORM without affect the business layer.

**3- technologies used at the frontend:**

- Reactjs

- Redux to manage the application state

- web pack for packaging

- axios for async calling

- redux-thunk

- enzyme , mocha , expect for testing

**4- Rest Service**

**- LogRequestAndResponseHandler** to log each request and response to log file

- end points

a- DocumentController

**-** DownloadDocument(string Id)

Endpoint for downloading the document by documentId

- GetAllDocuments()

Get all documents metadata for each user

- UploadDocument()

Upload the document to the server

b- AccountController: for authentication