**Design document**

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

- language : C#

-database: Microsoft SQL Server

- ORM: Entity framework

- DI Container: Ninject

- Architecture: N Tier

- Logger: log4net

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

 - Strategy with factory pattern:

to be able to extend any new download strategy in the feature without edit the currently implemented strategies. To add new strategy just implement IDownloadStrategy interface

and add the new added class at the Nninject service register;

-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.

**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**

- Technology used:

Web API

-end points

a- DownloadController

**-** DownloadBatchFiles(BatchSources batchSources)

Endpoint for downloading the files from the sources

- GetReadyForProcessingFiles()

Get collections of downloaded files and ready for processing

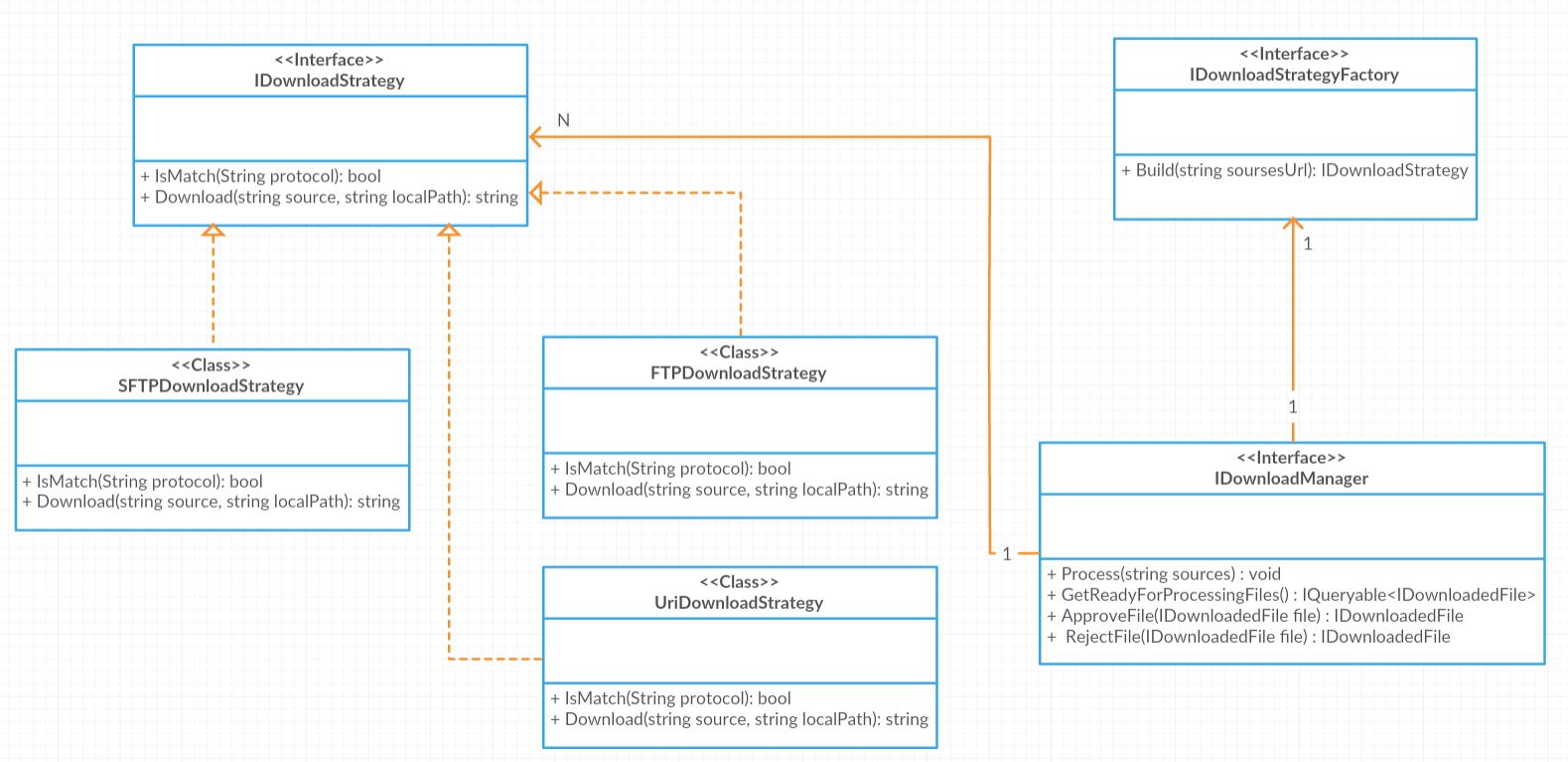
- ApproveFile(DownloadedFile file)

Update the file status to be approved

- RejectFile(DownloadedFile file)

Update the file status to be Rejected

**Class diagram:**



**Component diagram:**

