SOFTWARE DESIGN

Media Organiser

Contents

Software Architecture 2

UML (Class Diagram) 2

Entity Relationship Diagram 3

Technology 4

Technology Stack 4

Programming Language 4

Framework 4

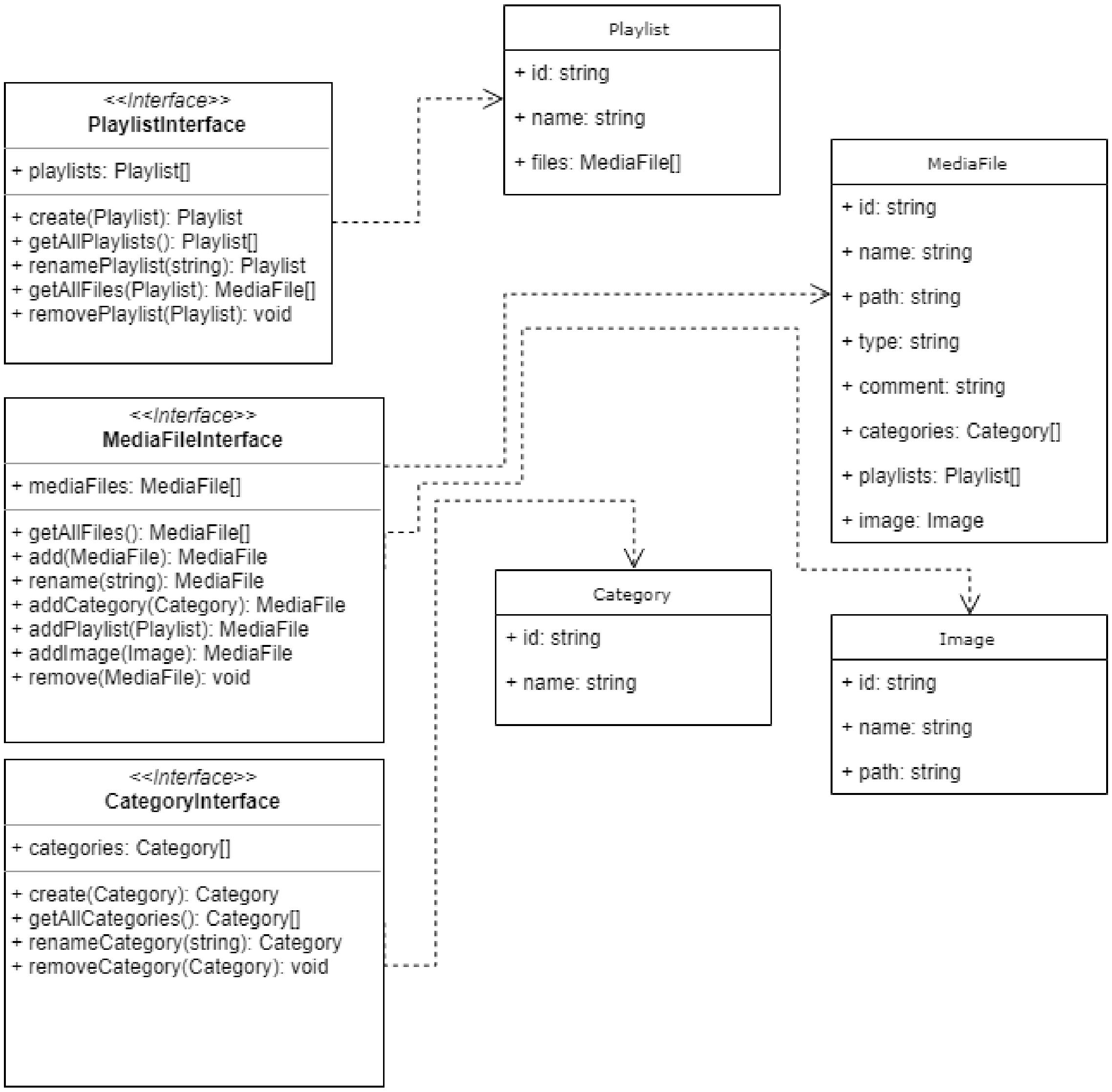
User Interface 5

Wireframes 5

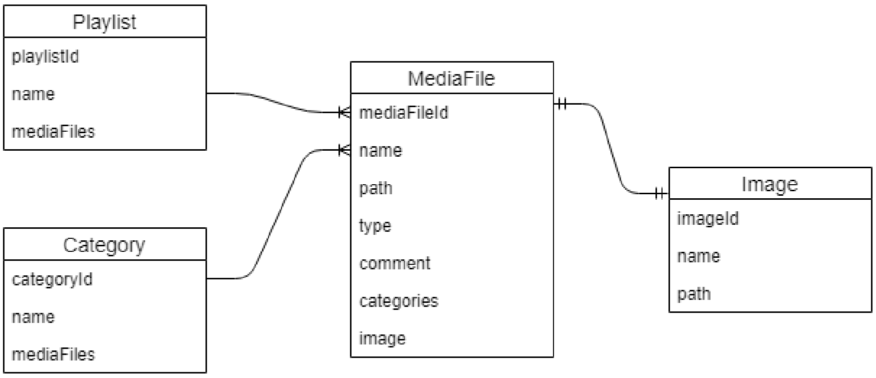
Components .... **Error! Bookmark not defined.**

Software Architecture

UML (Class Diagram)


Playlist
+ id: string
+ name: string
+ files: MediaFile[]
<<Interface>>
PlaylistInterface
+ playlists: Playlist[]
+ create(Playlist): Playlist
+ getAllPlaylists(): Playlist[]
+ renamePlaylist(string): Playlist
+ getAllFiles(Playlist): MediaFile[]
+ removePlaylist(Playlist): void
MediaFile
+ id: string
+ name: string
+ path: string
+ type: string
+ comment: string
+ categories: Category[]
+ playlists: Playlist[]
+ image: Image
<<Interface>>
MediaFileInterface
+ mediaFiles: MediaFile[]
+ getAllFiles(): MediaFile[]
+ add(MediaFile): MediaFile
+ rename(string): MediaFile
+ addCategory(Category): MediaFile
+ addPlaylist(Playlist): MediaFile
+ addImage(Image): MediaFile
+ remove(MediaFile): void
Category
+ id: string
+ name: string
Image
+ id: string
+ name: string
+ path: string
<<Interface>>
CategoryInterface
+ categories: Category[]
+ create(Category): Category
+ getAllCategories(): Category[]
+ renameCategory(string): Category
+ removeCategory(Category): void

Entity Relationship Diagram



Technology

Technology Stack

The requirement of this application encourages me to create a desktop application. This would mean the artefact created for this application should be executable on a desktop environment. To achieve this, some considerations for programming language and frameworks needs to be made in order to create an efficient application which is compatible with desktop environment.

The intention of the application development would be to create a web application and wrap it into an exe file where to make it interact with the local file system and also have all the rich features of a web application.

Programming Language

The following programming languages will be used in the application:

- TypeScript
- JavaScript
- HTML
- CSS

Framework

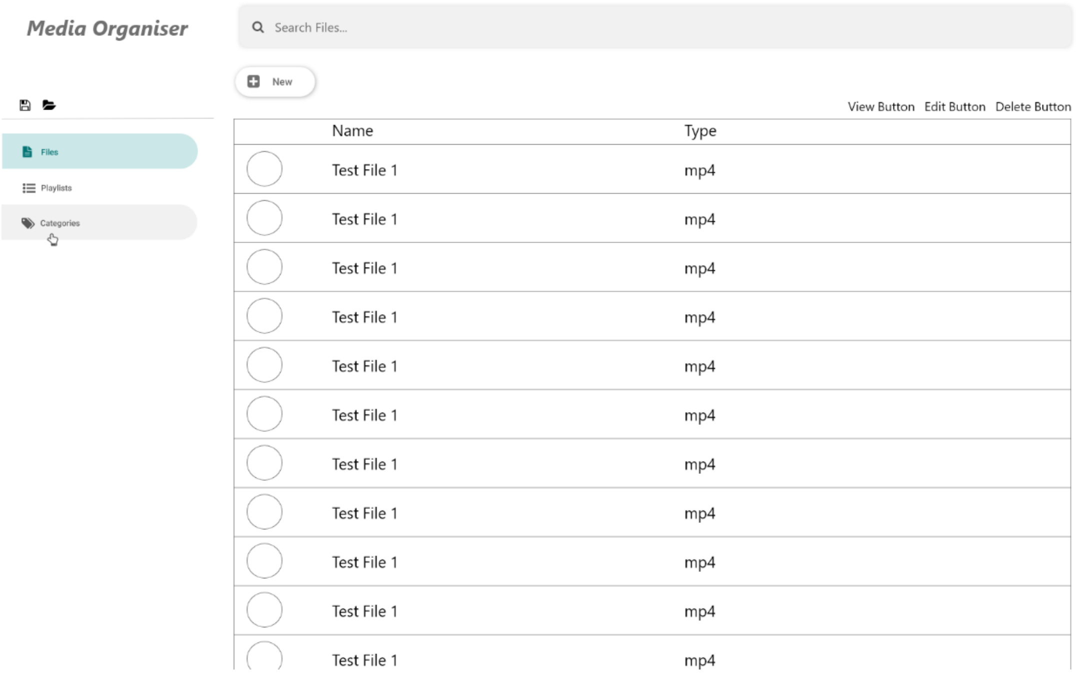
In order to accomplish the requirement and produce a robust and high-quality application, some considerations needs to be made about the frameworks that will be used for the development of this application.

To develop the web version, Angular (https://angular.io/) seems to be the perfect choice as it follows MVVM design pattern (Model-View-ViewModel). This gives the flexibility and robustness of managing the user interface and separating data from the views. Angular also ships with built-in packages which provides basic functionality out of the box (such as routing) and is easy to configure.

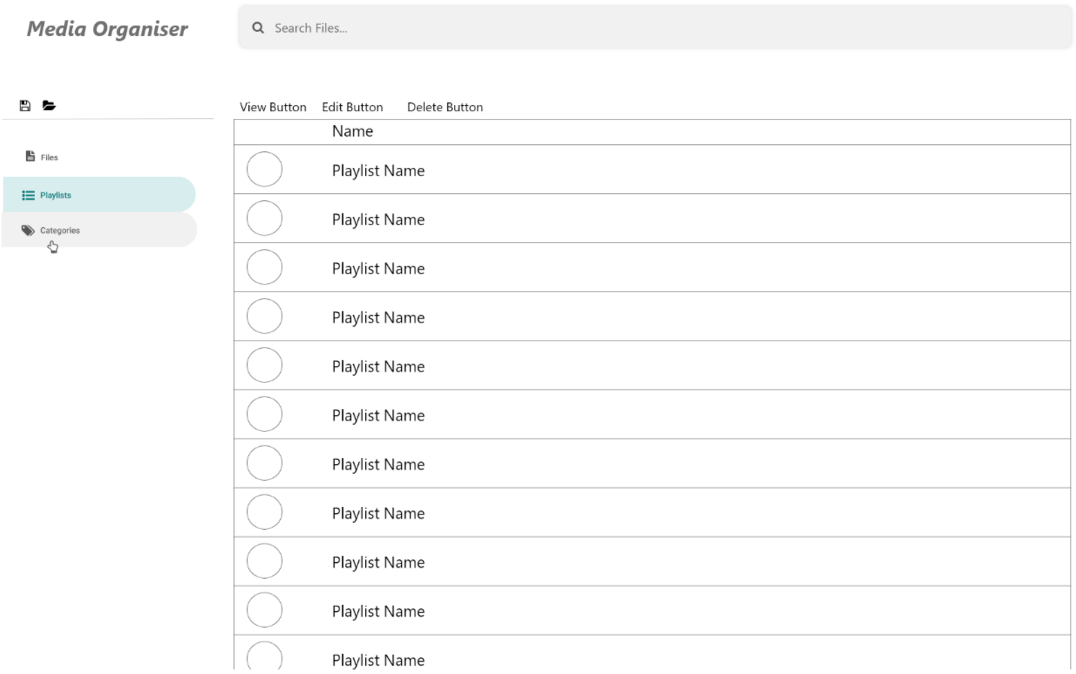
In order to wrap the web application into a desktop application to make it interact with user's local filesystem, Electron (https://electronjs.org/) seems to be the perfect choice. Electron provides a chromium engine which let the web application run as a native application. It also provides an API to interact with the filesystem and other OS functionality.

User Interface

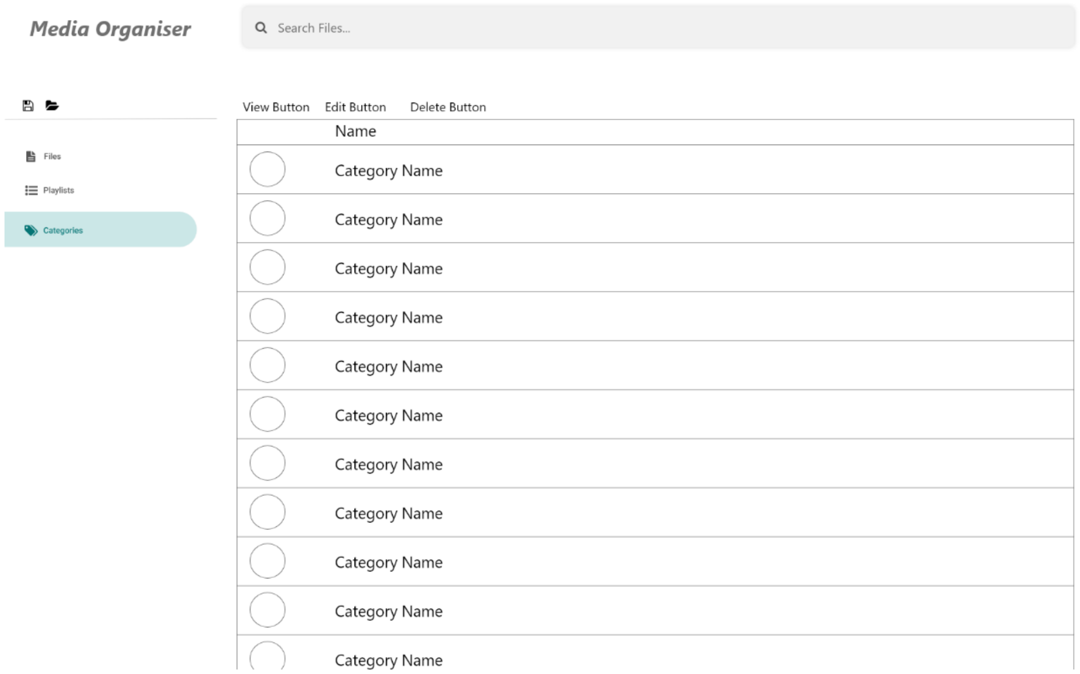
Wireframes



	Name	Type
○	Test File 1	mp4
○	Test File 1	mp4
○	Test File 1	mp4
○	Test File 1	mp4
○	Test File 1	mp4
○	Test File 1	mp4
○	Test File 1	mp4
○	Test File 1	mp4
○	Test File 1	mp4
○	Test File 1	mp4
○	Test File 1	mp4



	Name
○	Playlist Name
○	Playlist Name
○	Playlist Name
○	Playlist Name
○	Playlist Name
○	Playlist Name
○	Playlist Name
○	Playlist Name
○	Playlist Name
○	Playlist Name
○	Playlist Name



	Name
○	Category Name
○	Category Name
○	Category Name
○	Category Name
○	Category Name
○	Category Name
○	Category Name
○	Category Name
○	Category Name
○	Category Name
○	Category Name