Система является каталогом различных источников данных на домашнем компьютере. Поддерживаются следующие источники данных: фотографии, клипы, фильмы, документы, аудио записи. Каждый источник может быть сгруппирован в папке с определённым названием. У каждого источника может быть название и краткий комментарий. У пользователя должна быть возможность создавать папки с источниками, просматривать источники, добавлять, удалять, редактировать. Должен быть реализован поиск источников по слову в названии и описании. Каждый источник должен иметь маленькое preview для быстрого просмотра.

Постановка задачи

# Функционал программы

* Добавление, удаление, обновление, считывание источников и папок (CRUD)
* Хранение и доступ состояний системы в базе данных
* Просмотр папок
* Поиск источников
* Показ preview

Слои построения системы

* Client (слой представления (WPF GUI)
* Server (слой базы данных состояний и доступа к ним (asp.net webAPI)
* Shared (слой сущностей всей системы)

Сущности

# Shared

* Item (Абстрактное представление источника данных)

Описание:

ID (int)

Название (string)

Описание (string)

Реализации (классы):

Источник изображения (string)

* + Photo :(фотография)
  + Clip (клип)
  + Film (фильм)
  + Document (документ)
  + Audio (аудиозапись)
* Folder (папка)

Описание:

Id(int)

Название(string)

# Client

* ArchieveClient
  + Http client \_client
* FoldersService
  + ArchieveClient - http клиент
* ItemsService
  + ArchieveClient - http клиент
* ItemFactory
* ServerConnection
  + Событие изменения БД ContextChanged
* MainWindowViewModel (менеджер главного окна)
  + Items – массив источников
  + Folders – массив папок
  + SelectedItem – выбранный пользователем источник
  + SelectedFolder – выбранная пользователем папка
  + Команды MVVM
* SettingsWindowViewModel (менеджер окна настроек)
  + Host
  + Port

# Server

* ItemController – менеджер источников на сервере
* FolderController – менеджер папок на сервере

USE CASES

1. ItemController (сервер)
   1. Создать источник Post()
   2. Удалить источник Delete()
   3. Обновить источник Put()
   4. Найти источник по ID GetById()
   5. Найти источник по строке поиска GetByString()
   6. Получить список источников GetAll()
2. FolderController (сервер)
   1. Создать папку Post()
   2. Удалить папку Delete()
   3. Обновить папку Put()
   4. Найти папку по ID GetById()
   5. Получить список папок GetAll()
3. ArchieveClient
   1. POST запрос на сервер
   2. DELETE запрос на сервер
   3. PUT запрос на сервер
   4. GET зарос на сервер
4. ItemsService
   1. Вызов методов ItemController на сервере с помощью ArchieveClient
5. FoldersService
   1. Вызов методов FolderController на сервере с помощью ArchieveClient
6. ItemFactory (порождающий паттерн)
   1. Создать источник на основе строки
7. MainWindowViewModel
   1. Вызов методов FoldersService и ItemsService
   2. Обновить элементы представления
   3. Установить привязку между моделями и элементами GUI