Data Access Layer

Models

Person
+ Id:int
+ PrimaryName:String
+ Job:String
+ DateOfBirth:DateTime
+ DateOfDeath:DateTime
+ NameRating:float

Movie
+ TitleID:String
+ TitleName:String
+ Rating:Rating
+ AverageRating:double

Bookmark
+ ID:int
+ titleID:String
+ userID:String
+ status:Bool

Rating
+ TitleID:String
+ UserID:int
+ Grade:int
+ ReviewText:String
+ RateDate:DateTime

Genres
+ ID:int
+ Name:String

Search
+ userID:int
+ searchString:String
+ searchDate:DateTime

TitleAkas
+ ID:String
+ ordering:int
+ title:String
+ region:String
+ attribute:String
+ type:String
+ language:String
+ isOriginalTitle:bool
+ Basics:List

TitleAkas
+ ID:String
+ type:String
+ isAdult:Bool
+ startYear:String
+ endYear:String
+ poster:String
+ description:String
+ rating:int
+ Akas:List

BookmarkService

<<IBookmarkService>>
Interface

+ GetBookmarks():IList
+ GetBookmarkById(int bookmarkId):Bookmark
+ CreateBookmark(int userId, string titleId):Bookmark
+ DeleteBookmark(Bookmark):Bool
+ DeleteBookmark(int bookmarkId):Bool

GenreService

<<IGenreService>>
Interface

+ GetGenre():IList
+ GetGenre(int id):Bookmark
+ CreateGenre(String name):Genres
+ DeleteGenre(Genres):Bool
+ DeleteGenre(int Id):Bool
+ UpdateGenre(int id, string name):Bool

MovieService

<<IMovieService>>
Interface

+ SearchMovies(int userId, String searchString):IList
+ GetSimilarMovies(String movieId)
+ CreateMovie(TitleBasics newMovie, TitleAkas newTitleAkas)
+ DeleteMovie(String movieId): Bool
+ UpdateMovie(String movieId, TitleBasics updatedMovie, TitleAkas updateTitleAkas): Bool

PersonService

<<IPersonService>>
Interface

+ GetPersons(): IList
+ GetPersonById(int Id): Person
+ GetPersonByName(String PrimaryName)
+ CreatePerson(String PrimaryName, DateTime DateOfBirth, DateTime DateOfDeath, float NameRating): Person
+ DeletePerson(int personId);

RatingService

<<IRatingService>>
Interface

+ GetRatingHistory():IList
+ GetRatingHistoryByUserId(int userID):IList
+ CreateRating(String titleID, int userID, int grade, String reviewText, String rateDate):Rating
+ ReadRatingsForMovie(String titleID):IList
+ UpdateRating(String titleID, int userID, int newRatingValue, string newReviewText, String rateDate):Void
+ DeleteRating(String titleID, int userID):Void

SearchService

<<ISearchService>>
Interface

+ GetSearchHistory():IList
+ GetSearchHistoryById(int _userID):List
+ CreateSearch(int userId, string searchString):Search
+ DeleteSearch(string searchString, int _userID):Bool

UserService

<<IUserService>>
Interface

+ GetUsers():IList
+ GetSimilarMovies(String movieId)
+ CreateMovie(TitleBasics newMovie, TitleAkas newTitleAkas)
+ DeleteMovie(String movieId): Bool
+ UpdateMovie(String movieId, TitleBasics updatedMovie, TitleAkas updateTitleAkas): Bool

---

Web Service Layer

BookmarkController
- _dataService:IGenreService

+ GetBookmark(int):Bookmark
+ CreateBookmark(CreateBookmarkModel):Bookmark
+ UpdateBookmark(int, bookmark):bookmark
+ DeleteBookmark(int):bool

GenresController
- _dataService:IGenreService

+ GetGenre(int):Genre
+ CreateGenre(CreateGenreModel):Genre
+ UpdateGenre(int, genre):genre
+ DeleteGenre(int):bool

MoviesController
- _movieService:IMovieService

+ SearchMovies(int, String):IList
+ GetSimilarMovies(String):IList
+ CreateMovie(MovieCreationModel):Movie
+ DeleteMovie(String):bool

PersonsController
- _personService:IPersonService

- Paging(IEnumerable, int, int, int string):Object
+ GetPersons(int, int):Person
+ GetPersonById(String):Person
+ GetPersonByName(String):Person

RatingController
- _ratingService:IRatingService

+ GetRatingHistory():IList
+ GetRatingHistoryByIUserId(int):IList
+ CreateRating(CreateRatingModel):Rating
+ UpdateRating(string, int):Rating
+ DeleteRating(String, Int):bool

SearchController
- _searchService:ISearchService

+ GetSearchHistory():IList
+ GetSearchHistoryById(int):IList
+ CreateSearch(SearchCreationModel):Search
+ DeleteSearch(SearchDeleteModel):bool

UserController
- _userService:IUserservice
- _secretKey:String

+ GetUserHistory():IList
+ Register(CreateUserModel):User
+ Login(UserModel)
+ GenereateJwtToken(User)

Models

Program
+ builder:var
+ app:var

+ AddAutentication(options):Services
+ AddAuthorization():Services
+ AddControllers():Services
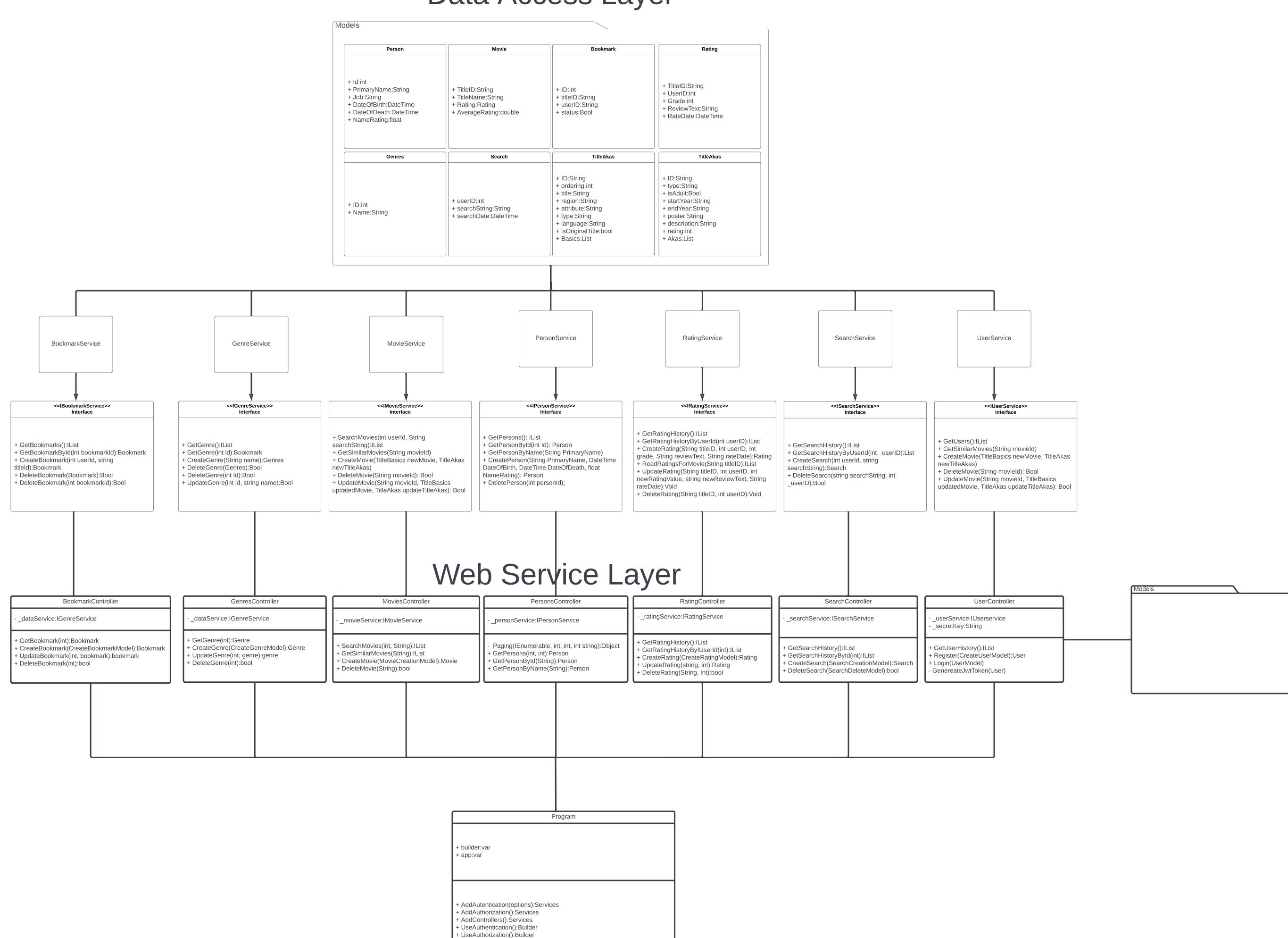+ UseAuthentication():Builder
+ UseAuthorization():Builder
+ MapControllers():Builder
+ Run():builder