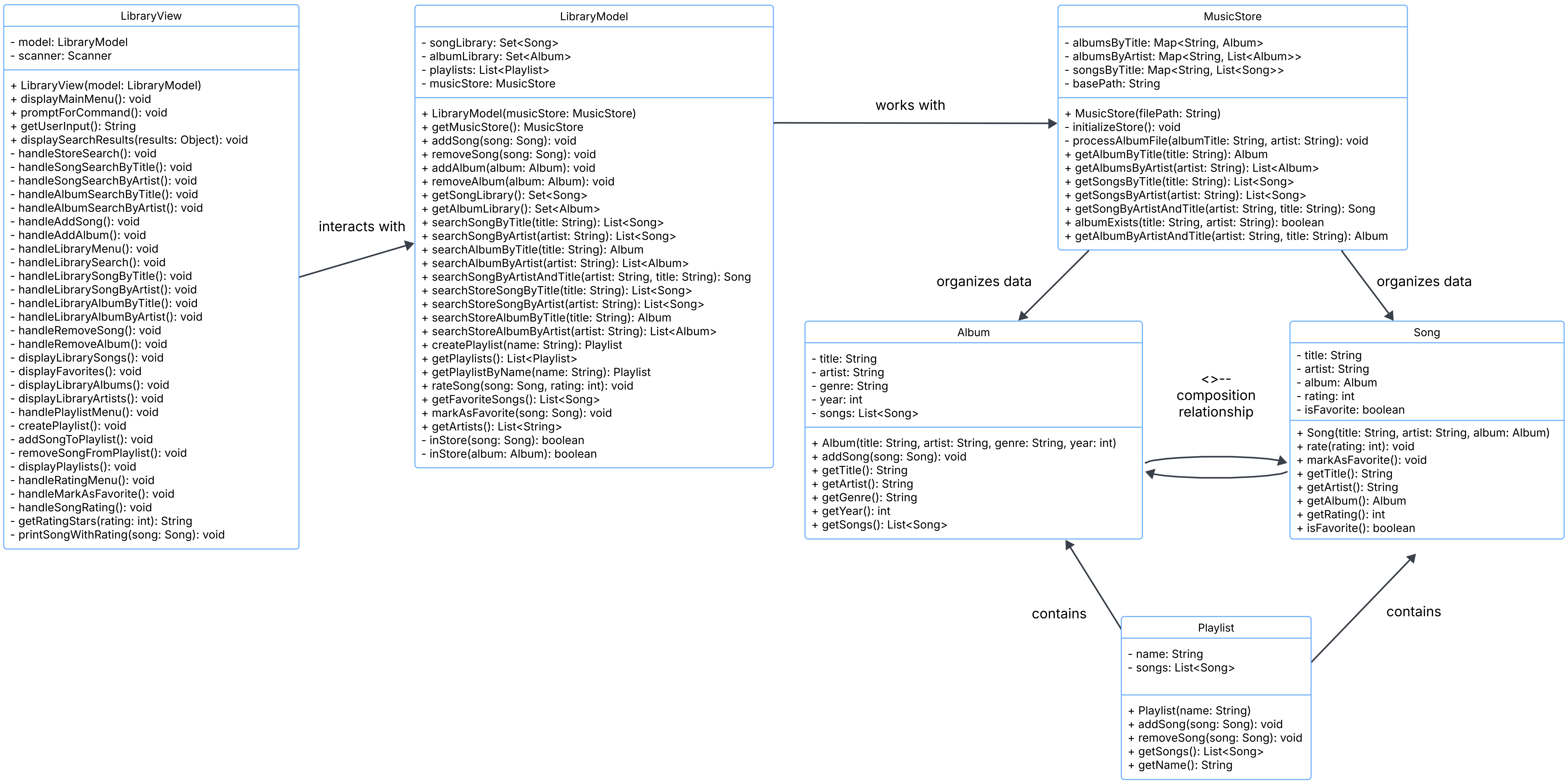

LibraryView

- model: LibraryModel
- scanner: Scanner

+ LibraryView(model: LibraryModel)
+ displayMainMenu(): void
+ promptForCommand(): void
+ getUserInput(): String
+ displaySearchResults(results: Object): void
- handleStoreSearch(): void
- handleSongSearchByTitle(): void
- handleSongSearchByArtist(): void
- handleAlbumSearchByTitle(): void
- handleAlbumSearchByArtist(): void
- handleAddSong(): void
- handleAddAlbum(): void
- handleLibraryMenu(): void
- handleLibrarySearch(): void
- handleLibrarySongByTitle(): void
- handleLibrarySongByArtist(): void
- handleLibraryAlbumByTitle(): void
- handleLibraryAlbumByArtist(): void
- handleRemoveSong(): void
- handleRemoveAlbum(): void
- displayLibrarySongs(): void
- displayFavorites(): void
- displayLibraryAlbums(): void
- displayLibraryArtists(): void
- handlePlaylistMenu(): void
- createPlaylist(): void
- addSongToPlaylist(): void
- removeSongFromPlaylist(): void
- displayPlaylists(): void
- handleRatingMenu(): void
- handleMarkAsFavorite(): void
- handleSongRating(): void
- getRatingStars(rating: int): String
- printSongWithRating(song: Song): void

LibraryModel

- songLibrary: Set<Song>
- albumLibrary: Set<Album>
- playlists: List<Playlist>
- musicStore: MusicStore

+ LibraryModel(musicStore: MusicStore)
+ getMusicStore(): MusicStore
+ addSong(song: Song): void
+ removeSong(song: Song): void
+ addAlbum(album: Album): void
+ removeAlbum(album: Album): void
+ getSongLibrary(): Set<Song>
+ getAlbumLibrary(): Set<Album>
+ searchSongByTitle(title: String): List<Song>
+ searchSongByArtist(artist: String): List<Song>
+ searchAlbumByTitle(title: String): Album
+ searchAlbumByArtist(artist: String): List<Album>
+ searchSongByArtistAndTitle(artist: String, title: String): Song
+ searchStoreSongByTitle(title: String): List<Song>
+ searchStoreSongByArtist(artist: String): List<Song>
+ searchStoreAlbumByTitle(title: String): Album
+ searchStoreAlbumByArtist(artist: String): List<Album>
+ createPlaylist(name: String): Playlist
+ getPlaylists(): List<Playlist>
+ getPlaylistByName(name: String): Playlist
+ rateSong(song: Song, rating: int): void
+ getFavoriteSongs(): List<Song>
+ markAsFavorite(song: Song): void
+ getArtists(): List<String>
- inStore(song: Song): boolean
- inStore(album: Album): boolean

MusicStore

- albumsByTitle: Map<String, Album>
- albumsByArtist: Map<String, List<Album>>
- songsByTitle: Map<String, List<Song>>
- basePath: String

+ MusicStore(filePath: String)
- initializeStore(): void
- processAlbumFile(albumTitle: String, artist: String): void
+ getAlbumByTitle(title: String): Album
+ getAlbumsByArtist(artist: String): List<Album>
+ getSongsByTitle(title: String): List<Song>
+ getSongsByArtist(artist: String): List<Song>
+ getSongByArtistAndTitle(artist: String, title: String): Song
+ albumExists(title: String, artist: String): boolean
+ getAlbumByArtistAndTitle(artist: String, title: String): Album

Album

- title: String
- artist: String
- genre: String
- year: int
- songs: List<Song>

+ Album(title: String, artist: String, genre: String, year: int)
+ addSong(song: Song): void
+ getTitle(): String
+ getArtist(): String
+ getGenre(): String
+ getYear(): int
+ getSongs(): List<Song>

Song

- title: String
- artist: String
- album: Album
- rating: int
- isFavorite: boolean

+ Song(title: String, artist: String, album: Album)
+ rate(rating: int): void
+ markAsFavorite(): void
+ getTitle(): String
+ getArtist(): String
+ getAlbum(): Album
+ getRating(): int
+ isFavorite(): boolean

Playlist

- name: String
- songs: List<Song>

+ Playlist(name: String)
+ addSong(song: Song): void
+ removeSong(song: Song): void
+ getSongs(): List<Song>
+ getName(): String