

# Table of Contents

1. Requirements
  2. Workplan
  3. Use Cases
  4. Use Cases Diagram
  5. Domain Model Diagram
  6. Add Album Requirements-level Diagram
  7. Add Album Desktop Implementation-level Diagram
  8. Add Album Mobile Implementation-level Diagram
  9. Add Album Web Implementation-level Diagram
- 

## Requirements

Each requirement has an associated ID. Each ID has two initial letters referring to the highest-level category (below functional and non-functional) it belongs to. Where appropriate, some IDs have a middle single letter referring to a subcategory. All IDs then have a two-digit number which distinguishes it from the other IDs which share its aforementioned letters. Periods separate each of these three components.

Priority numbers are given before each requirement ID. Priority is given on a scale from 1 to 3, where 1 means highest priority. Generally, a priority 1 element will be something that is present in the Project Overview. A priority 2 element will be something that adds an important functionality, while a priority of 3 suggests that the requirement is not necessary.

## Functional

### Music Library Management

#### Albums

- [1] [ML.A.01] The home audio system shall allow the user to add an album by inputting the text of the album title (mandatory) and its artist's name (mandatory), its genre (optional), its release date (optional). When inputting each album, the system shall allow the user to add sequential list of at least one song, each with its own title and duration. To add a single song that belongs to an album which hasn't been added yet, the system shall allow the user to add an album comprised of only one song.
- [3] [ML.A.02] The home audio system shall allow the user to add songs to an album after the album has been created.
- [2] [ML.A.03] The home audio system shall allow the user to remove an album (and by extension, all the songs associated with it) from the library.
- [2] [ML.A.04] The home audio system shall allow the user to remove individual songs from the library.

## Playlists

- [1] [ML.P.01] The home audio system shall allow the user to create a playlist by selecting songs that already exist in the library.
- [3] [ML.P.02] The home audio system shall allow the user to change the order of songs already added to a playlist.
- [2] [ML.P.03] The home audio system shall allow the user to remove a playlist from the library, which should not delete the songs contained in the process.
- [2] [ML.P.04] The home audio system shall allow the user to add songs from the library to an existing playlist.
- [2] [ML.P.05] The home audio system shall allow the user to remove songs from an existing playlist.

## User Interface

- [1] [ML.U.01] The home audio system shall allow the user to see all songs in its library sorted by album name or artist name.
- [2] [ML.U.02] The home audio system shall allow the user to see all songs in its library sorted by song title.
- [1] [ML.U.03] The home audio system shall allow the user to see all the songs in a given album or playlist.
- [1] [ML.U.04] The home audio system shall allow the user to see the artist, genre, and release date of an album.
- [1] [ML.U.04] The home audio system shall allow the user to see the duration, title, and album of each song.

## Home Location Management

- [1] [HL.01] The home audio system shall allow the user to add home locations by inputting the text of the location's name.
- [2] [HL.02] The home audio system shall allow the user to remove existing home locations.
- [1] [HL.03] The home audio system shall allow the user to add songs, albums, and playlists to a queue to play next at each home location.
- [2] [HL.04] The home audio system shall allow the user to remove individual songs from the queue at each home location.

## Volume Control

- [1] [HL.V.01] The home audio system shall allow the user to increase and decrease the speaker volume at each home location.
- [1] [HL.V.02] The home audio system shall allow the user to temporarily mute the speaker volume at each home location.

## User Interface

- [2] [HL.U.01] The home audio system shall allow the user to see the current volume level of each

home location.

- [2] [HL.U.02] The home audio system shall allow the user to see each home location has been temporarily muted.
- [1] [HL.U.03] The home audio system shall allow the user to see the name of each home location currently added to the system.
- [2] [HL.U.04] The home audio system shall allow the user to see which songs have been queued up to a particular home location.

## Playback

- [2] [PB.01] The home audio system shall keep track of the current playback position, to the nearest second, relative to the start of the song, for each song playing in each home location.
- [2] [PB.02] The home audio system shall allow the user to pause and resume the currently playing song at each location.
- [2] [PB.03] The home audio system shall allow the user to skip ahead to the next song in the queue while a song is paused or is playing.

## Jumping Playback Positions

- [1] [PB.J.01] When the currently playing song finishes, the next song in the queue shall start playing. If no songs are left in the queue, playback shall stop.
- [2] [PB.J.02] The home audio system shall allow the user to jump back to the beginning of the currently playing song, if the user initiates the command when current playback position is five seconds or more after the song start.
- [2] [PB.J.03] The home audio system shall allow the user to skip back to the previously played song, if the user initiates the command when current playback position is less than five seconds after the current song start.

## User Interface

- [2] [PB.U.01] The home audio system shall allow the user to see the song currently playing in each home location, along with this song's current playback position, its title, its album's title, its artist's title, and its duration.

# Non-Functional

## Interoperability

- [1] [IN.01] The user may interact with the home audio system via three different applications: one for laptop/desktop machines, one for mobile devices, and one for web browsers.

## Laptop/Desktop Application

- [1] [IN.L.01] The laptop/desktop application must be implemented in Java.
- [1] [IN.L.02] The laptop/desktop application user interface must be implemented with the Java Swing framework.

## Mobile Application

- [1] [IN.M.01] The mobile application must be implemented in Java on the Android platform.
- [1] [IN.M.02] The mobile application user interface must be implemented using the UI development framework that comes with Android Studio.

## Web Application

- [1] [IN.W.01] The web application must be primarily written in PHP.

## Speed

- [3] [SP.01] For the addition of a home location, album, or playlist: The time between when the user submits a correctly filled-out form and the time when the entity is added to the system should be one second or less.

## Persistence

- [1] [PS.01] Application data must be saved in text files (e.g., XML) or in a database.
  - [1] [PS.02] Changes to application data made on one application shall not be recognized by the other applications.
- 

# Workplan

Each item should be completed between the previous due date and its actual due date, thus indicating its approximate level of effort required. Lowest priority items should not be developed past April 8.

## Already done:

- Add Album [ML.A.01]

## Due Feb. 26

### Descriptions

- Architecture description -- Evan
- Architecture block diagram -- Nicky

### Functionality and UI

- Create Playlist [ML.P.01] -- Eva Desktop, Thomas Web, Pierre Mobile
- Add Location [HL.01] -- Eva Desktop, Thomas Web, Pierre Mobile
- View Names of Locations -- Eva Desktop, Thomas Web, Pierre Mobile
- Add Song to Playlist [ML.P.04] -- Eva Desktop, Thomas Web, Pierre Mobile

## Test Writing

- Add Album Unit Test -- Nicky
- Create Playlist Unit Test -- Evan
- Add Location Unit Test -- Nicky
- Add Song to Playlist Unit Test -- Evan

## Due Mar. 4

### Descriptions

- Detailed design description -- Evan
- Detailed design class diagram -- Evan
- Work plan update -- Thomas

### Functionality and UI

- Assign Playlist to Location [HL.03] -- Nicky Desktop, Thomas Web, Eva Mobile
- Assign Album to Location [HL.03] -- Nicky Desktop, Thomas Web, Eva Mobile
- Assign Song to Location [HL.03] -- Nicky Desktop, Thomas Web, Eva Mobile
- Turn Volume Up or Down [HL.V.01] -- Nicky Desktop, Thomas Web, Eva Mobile
- Temporarily Mute Volume [HL.V.02] -- Nicky Desktop, Thomas Web, Eva Mobile
- View All Locations (currently playing song, volume level, current location in song) [PB.U.01] -- Nicky Desktop, Thomas Web, Eva Mobile

## Test Writing

- Assign Playlist, Album, Song to Location Unit Tests -- Evan
- Turn Volume Up or Down Unit Tests -- Pierre
- Temporarily Mute Volume Unit Tests -- Pierre

## Due Mar. 11

### Descriptions

- Unit testing -- Thomas
- Component testing

### Functionality

- Next Song Plays When Current Song Finishes [PB.J.01] -- Pierre Desktop, Thomas Web, Nicky Mobile
- Delete Playlist from Library [ML.P.03] -- Pierre Desktop, Thomas Web, Nicky Mobile
- Delete Album from Library [ML.A.03] -- Pierre Desktop, Thomas Web, Nicky Mobile
- Delete Songs from Library [ML.A.04] -- Pierre Desktop, Thomas Web, Nicky Mobile

## Test Writing

- Next Song Plays When Current Song Finishes Unit Tests -- Evan

- Delete Playlist, Album, Songs from Library Unit Tests -- Eva
- Write first component tests -- Eva

## Due Mar. 18

### Descriptions

- System testing -- Evan
- Performance/stress testing -- Evan
- Work plan update -- Thomas

### Functionality and UI

- Play/Pause [PB.02] -- Eva Desktop, Thomas Web, Pierre Mobile
- Skip Songs [PB.03] -- Eva Desktop, Thomas Web, Pierre Mobile
- Jump to Start/End of Song [PB.J.02], [PB.J.03] -- Eva Desktop, Thomas Web, Pierre Mobile

### Test Writing

- Continue writing component tests
- Write first system tests -- Nicky
- Write Play/Pause unit tests -- Thomas
- Write Skip Songs unit tests -- Thomas
- Write Jump to Start/End of Song unit tests -- Eva

## Due Mar. 25

### Descriptions

- Release pipeline -- Evan
- Work plan update -- Thomas

### Functionality and UI

- Remove Song from Playlist [ML.P.05] -- Nicky Desktop, Thomas Web, Eva Mobile
- Change Order of Songs in Playlist [ML.P.02] -- Nicky Desktop, Thomas Web, Eva Mobile
- Add Songs after Album Created [ML.A.02] -- Nicky Desktop, Thomas Web, Eva Mobile

### Test Writing

- Continue writing system tests -- Evan
- Begin writing performance/stress tests -- Eva
- Write Remove Song from Playlist unit tests -- Nicky
- Write Change Order of Songs in Playlist unit tests -- Nicky
- Write Add Songs after Album Created unit tests -- Pierre

## Due Apr. 1

## Functionality and UI

- View Volume Level of Location [HL.U.01] -- Pierre Desktop, Thomas Web, Nicky Mobile
- View Music Assigned to each Location [HL.U.04] -- Pierre Desktop, Thomas Web, Nicky Mobile

## Test Writing

- Continue writing performance/stress tests -- Evan
- Finish writing component, system, performance/stress tests -- Evan
- Finalize unit tests (should have been covered already) -- Pierre

## Due Apr. 8

- Draft Presentation -- All of us
- Practice Presentation as group -- All of us
- Slack time (in case features take more time than anticipated) -- All of us

## Due Apr. 15

- Prepare and Submit Source Code -- Evan
  - Slack time (in case features take more time than anticipated) -- All of us
- 

# Use Cases

## Title: Add Album

Actors: User

Requirements: [ML.A.01]

Main Scenario:

```
1: System shows the Add Album form
2: User enters the album title (mandatory) and its artist's name (mandatory), its
   genre (optional), its release date (optional), as well as a sequential list of
   songs.
3: System adds the album to the application
4: System shows the Add Album form
```

Alternatives:

```
3a: Either of the mandatory fields is empty or only spaces
3a1: System shows error message "Mandatory fields cannot be empty!"; use case
    continues at step 1.
3b: The sequential list of songs is empty or only spaces
3b1: System shows error message "List of songs must contain at least one song.";
    use case continues at step 1.
```

## Title: Add Home Location

Actors: User

Requirements: [HL.01]

Main Scenario:

```
1: System shows the Add Home Location form
2: User enters the location's name.
3: System adds the location to the application
4: System shows the Add Home Location form
```

Alternatives:

```
3a: Location name is empty or only spaces
3a1: System shows error message "Location name cannot be empty!"; use case
continues at step 1.
```

## Title: Add Song

Actors: User

Requirements: [ML.A.02]

Main Scenario:

```
1: System shows the Add Song form
2: User enters the song title (mandatory), a song file and chooses from a drop-down
list an album to which the song will be added
3: System adds the song to the application
4: System shows the Add Song form
```

Alternatives:

```
3a: Song title is empty or only spaces
3a1: System shows error message "Song title cannot be empty!"; use case continues
at step 1.
3b: The song file is empty or only spaces
3b1: System shows error message "Song file cannot be empty."; use case continues at
step 1.
```

## Title: Create PPlaylist



**Actors:** User

**Requirements:** [ML.P.01]

**Main Scenario:**

```
1: System shows the Create Playlist form
2: User enters the playlist name, and chooses songs from a list of all songs in the system
3: System adds the playlist to the application
4: System shows the Create Playlist form
```

**Alternatives:**

```
3a: Playlist name is empty or only spaces
3a1: System shows error message "Playlist name cannot be empty!"; use case continues at step 1.
3b: No song is selected
3b1: System shows error message "To create a playlist, at least one song must be selected."; use case continues at step 1.
```

## Title: Playback

**Actors:** User

**Requirements:** [HL.V.01], [HL.V.02], [PB.02]

**Main Scenario:**

```
1: System shows the Playback form
2: User chooses an existing home location, and sets a volume, mutes, unmutes, pauses playback or resumes playback.
3: System modifies the playback parameters of the desired home location
4: System shows the Playback form
```

**Alternatives:**

```
3a: No location is selected
3a1: System shows error message "A home location must be selected."; use case continues at step 1.
3b: No action is selected
3b1: System shows error message "One action must be selected."; use case continues at step 1.
3c: More than one action is selected
3c1: System shows error message "Only one action can be selected."; use case continues at step 1.
```

# Title: Play Music

Actors: User

Requirements: [HL.03]

## Main Scenario:

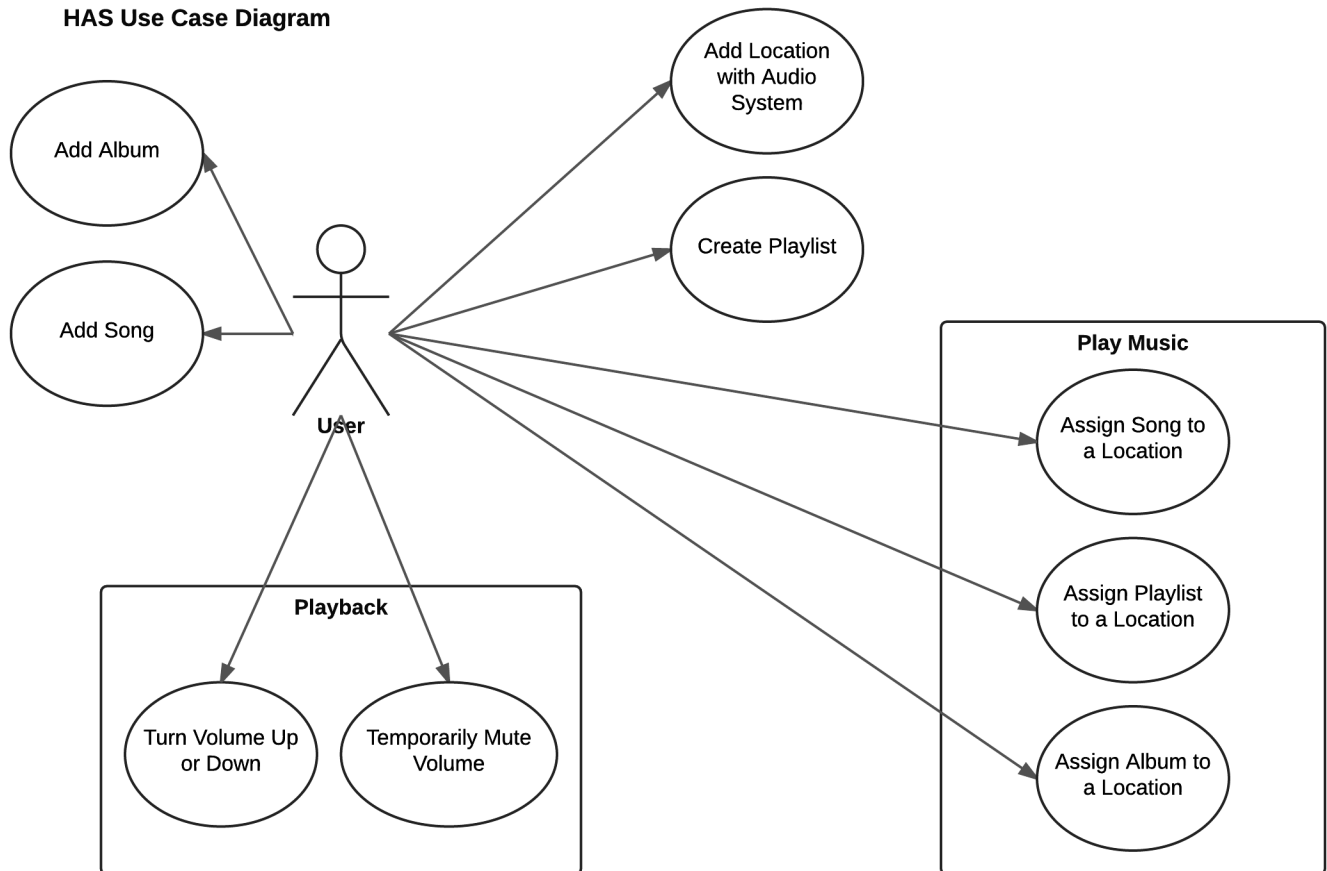
```
1: System shows the Play Music form
2: User chooses an existing home location, as well as a song, an album and/or a
playlist from separate drop-down lists.
3: System adds the selected media to the playback queue of the desired home
location
4: System shows the Play Music form
```

## Alternatives:

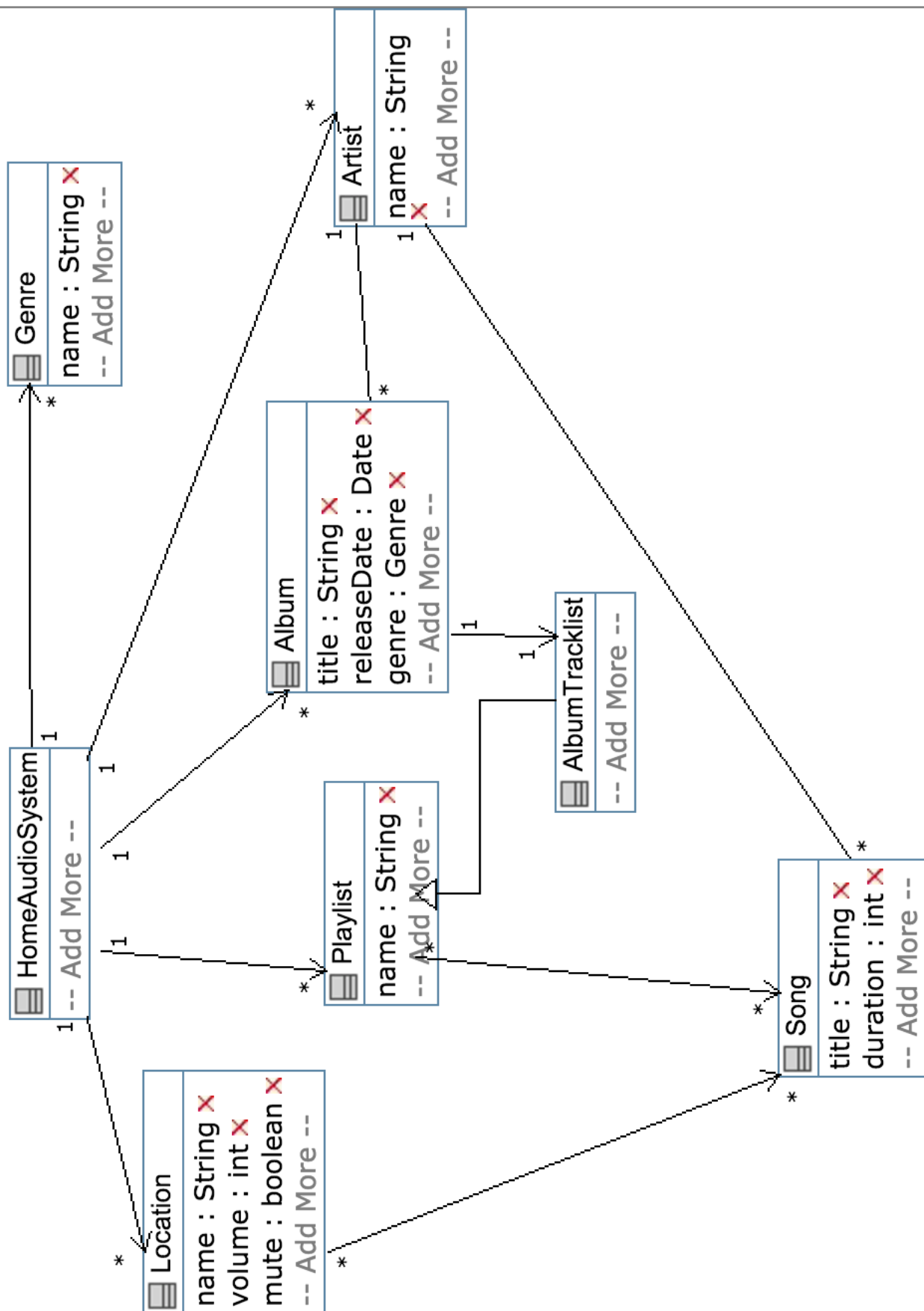
```
3a: No location is selected
3a1: System shows error message "A home location must be selected."; use case
continues at step 1.
3b: No media is selected
3b1: System shows error message "At least one song, album or playlist must be
selected"; use case continues at step 1.
```

# Use Cases Diagram

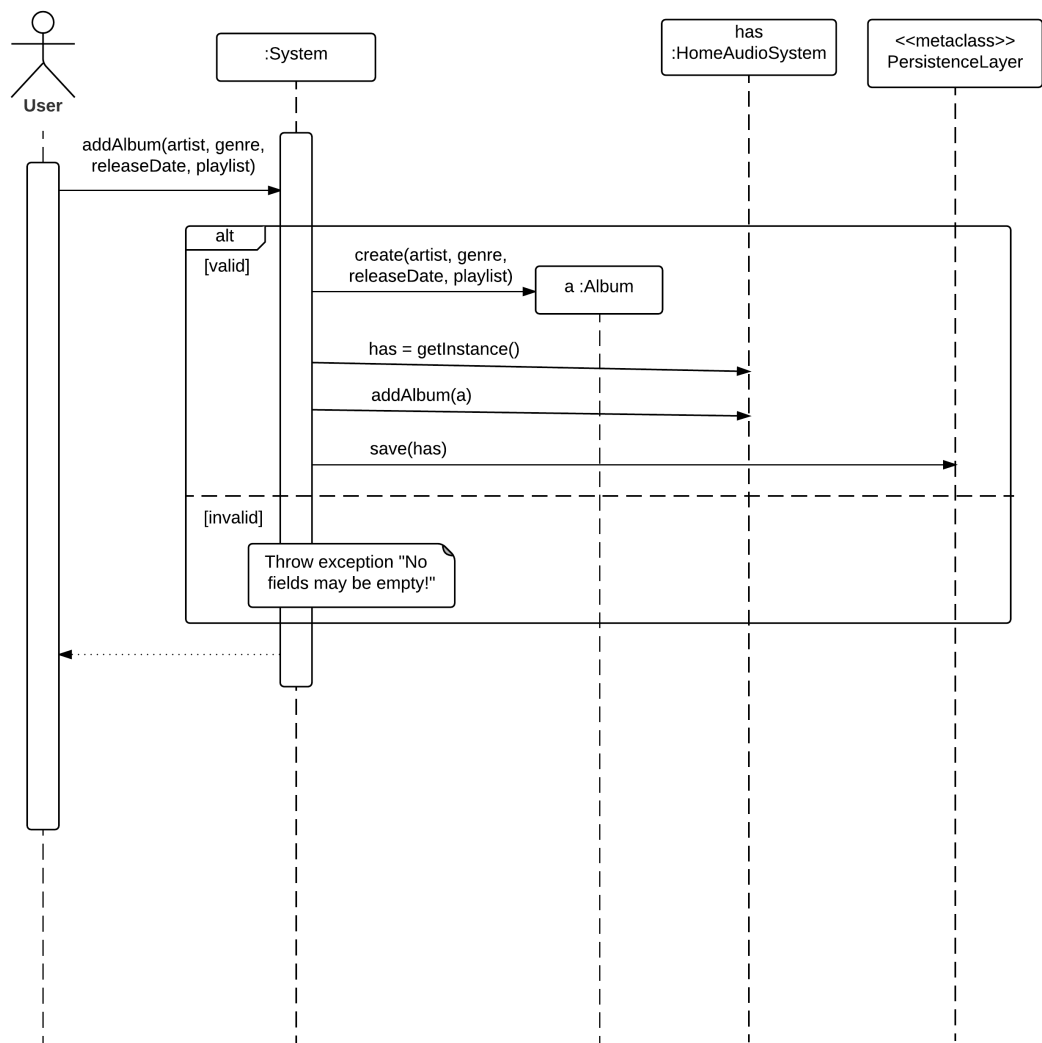
HAS Use Case Diagram



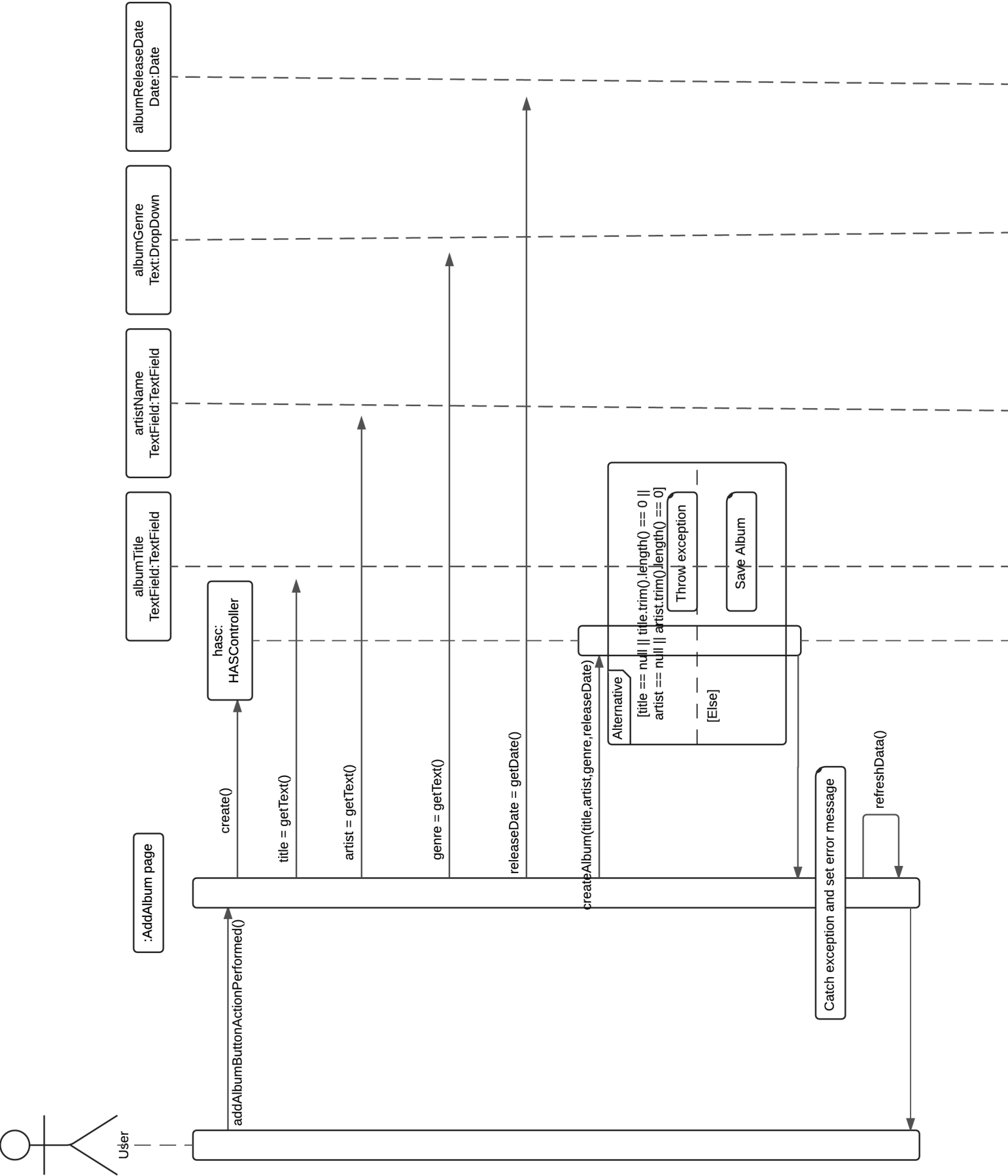
## Domain Model Diagram



# Add Album Requirements-level Sequence Diagram



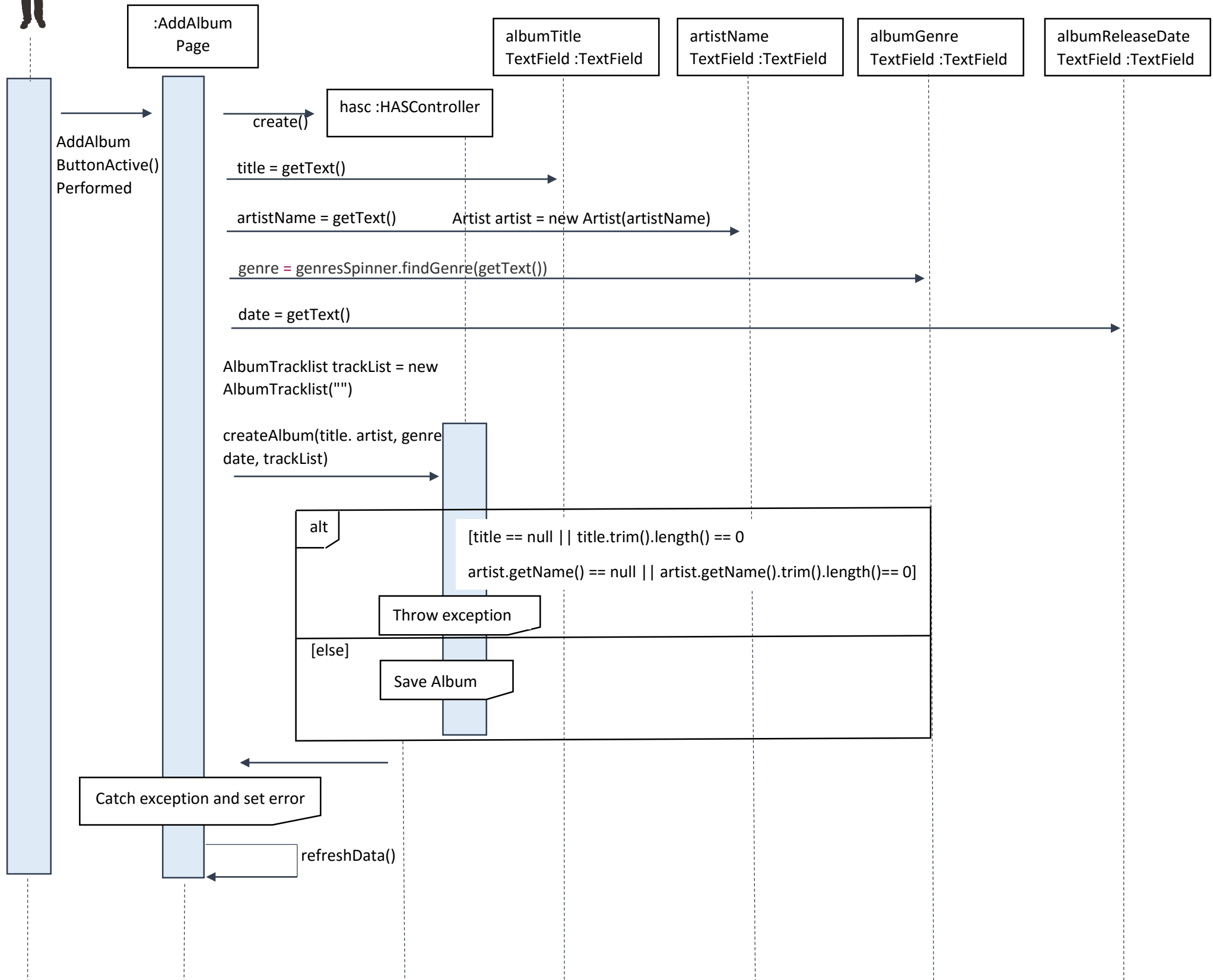
Add Album Desktop Implementation-level Sequence Diagram





User

## Add Album Mobile Implementation-level Sequence Diagram



Add Album Web Implementation-level Sequence Diagram

