



# McGill

**ECSE 321**

**Introduction to Software Engineering**

**Home Audio System**

**Deliverable 1**

**February 22 2016**

**Kristina Pearkes 260 520 832**

**Aidan Piwowar 260 625 505**

**Kareem Halabi 260 616 162**

**Aurelie Pluche 260 622 575**

**Alexander Orzechowski 260 610 696**

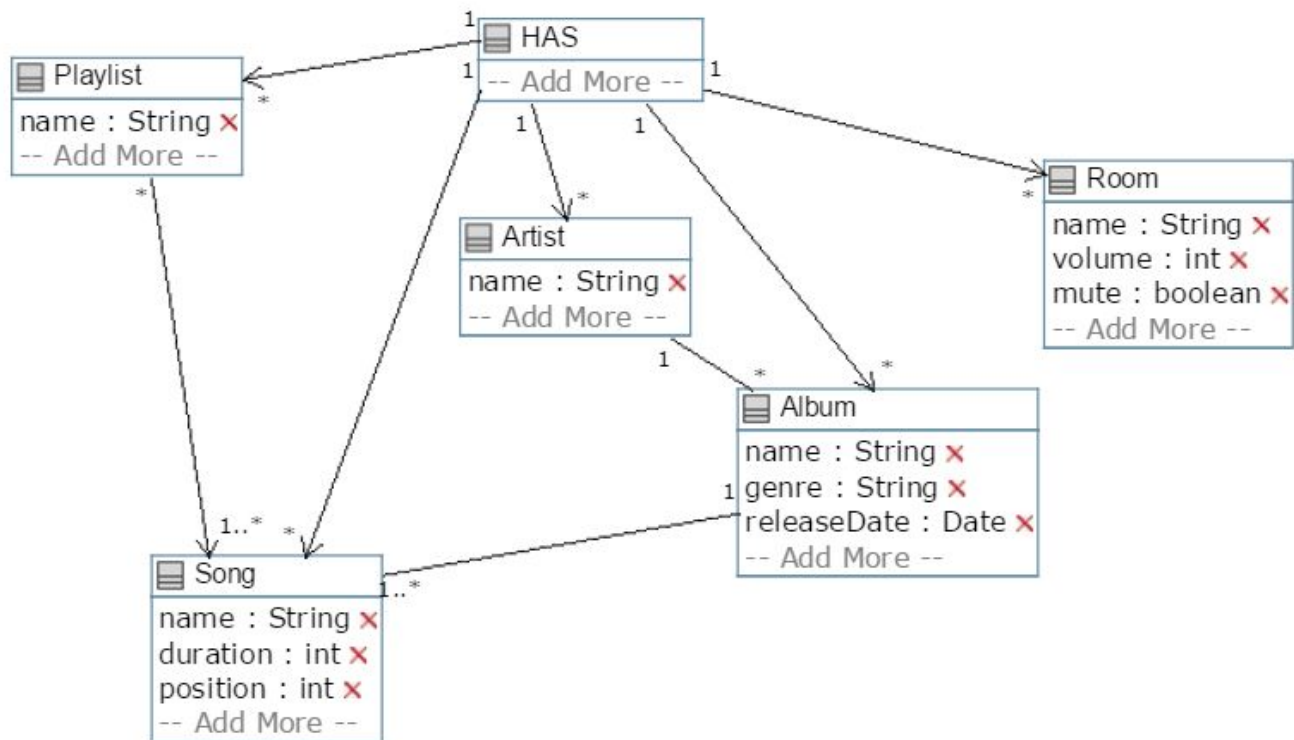
# System Requirements

## Current Umlle Model:

<https://cruise.eecs.uottawa.ca/umpleonline/umple.php?model=160213633256>

The HAS singleton manages all the components of the model including Playlists, Songs, Artists, Albums and Rooms. A Playlist has a name and at least one Song associated with it. One Song can belong to any number of Playlists, but it does not need to know which Playlists it belongs to. HAS also manages multiple Rooms with different names and keeps track of the volumes and mute statuses in each of the Rooms. The music itself is comprised of three components:

1. An Artist has a name and 0 to many Albums.
2. Albums which belong to 1 Artist, have a name, genre, release date and at least 1 Song.
3. A Song which has a name, duration, album position and belongs to 1 Album as well as zero to many playlists.



## **Intro**

### **Functional Requirements (sorted by highest priority at the top)**

1. The system shall allow the user to add artists. (Req-2.3)
2. The system shall allow the user to add albums. (Req-2.1)
3. The system shall allow the user to associate an album to an artist. (Req-3.1)
4. The system shall allow the user to add songs to an album. (Req-2.2)
5. The system shall allow the user to add a room (Req-1.2)
6. The system shall function in multiple rooms. (Req-1.3)
7. The system shall allow the user to create playlists. (Req-2.4)
8. The system shall provide the ability to stream different items to various locations. (Req-1.1)
9. The system shall organize music by artist or album. (Req-3.2)
10. The system shall provide the ability to set the volume in each location. (Req-1.4)
11. The system shall provide the ability the ability to temporarily mute music in each location. (Req-1.5)

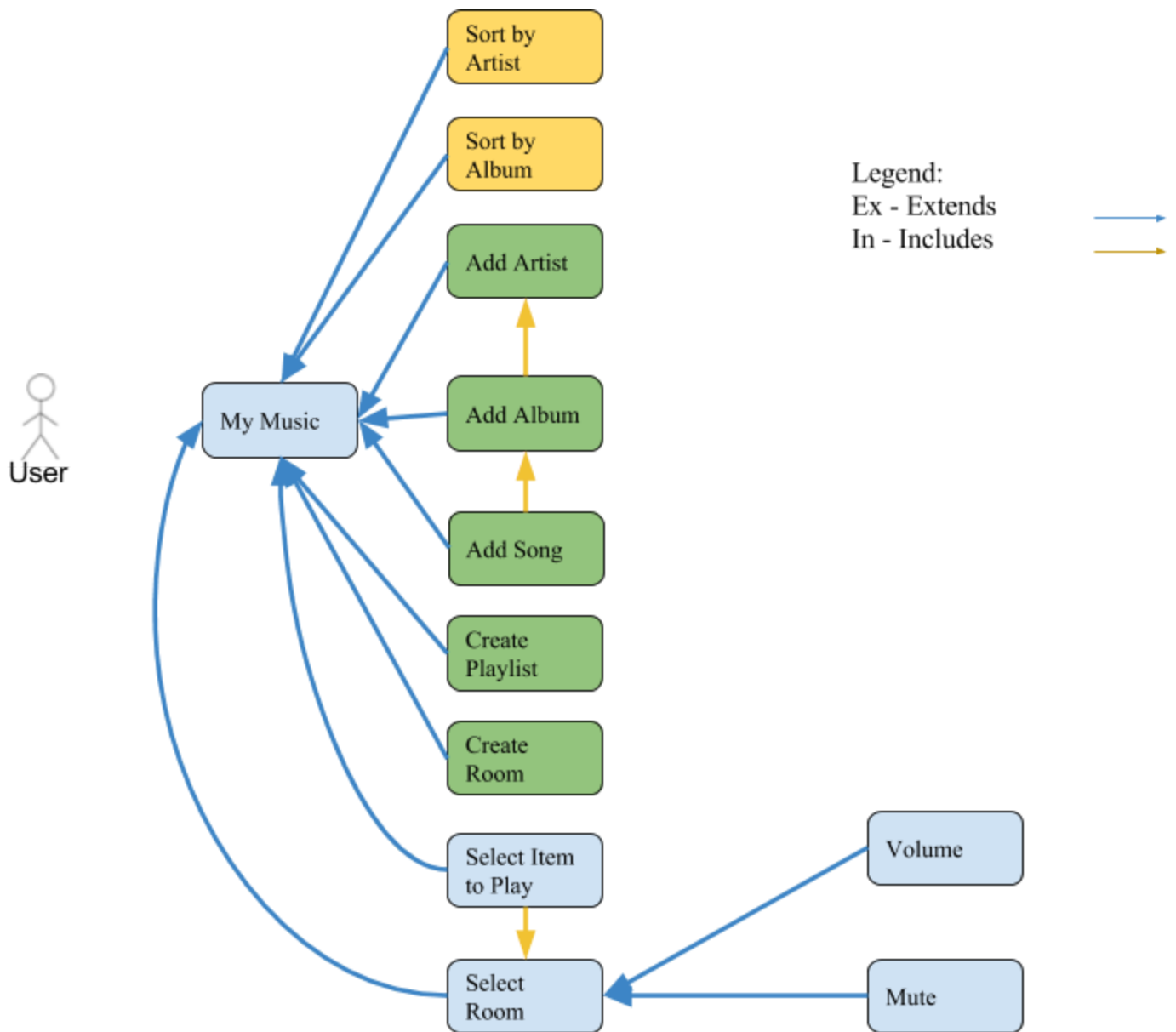
### **Non-Functional Requirements**

1. The system shall operate on the desktop, Android, and web platforms. (Req-4.1)

### **Work Plan**

1. The system shall limit the maximum volume. (Req-4.2)
- Deliverable 2 by March 7th (Medium effort):
    - Add a room (1.2)
    - Function in multiple rooms (1.3)
    - Create playlist(2.4)
    - Select an item to play (1.1)
    - Description of architecture of proposed solution including a block diagram
    - Description of detailed design of proposed solution including class diagram
  - Deliverable 3 by March 21st (High effort):
    - Organize music by album or artist (3.2)
    - Choose a volume level for a room (1.4)
    - Be able to mute a room(1.5)
    - Description of unit testing
    - Description of component testing
    - Description of system testing
    - Description of performance/stress testing
  - Deliverable 4 by March 28th (low)
    - Description of release pipeline.
  - Deliverable 5 by April 11/12/14 (Medium)
    - Presentation about project
  - Deliverable 6 by April 15 (Medium-High)
    - Source code of full implementation on each supported platform

## Use Case Diagram



### Use Cases:

**Title:** Sort by Artist

**Actor:** User

**Requirements:** Req-3.2

### Main Scenario:

1. User selects to sort songs alphabetically by artist
2. System sorts songs alphabetically by artist
3. System displays songs sorted alphabetically by artist

**Title:** Sort by Album

**Actor:** User

**Requirements:** Req-3.2

**Main Scenario:**

1. User selects to sort songs alphabetically by album
2. System sorts songs alphabetically by album
3. System displays songs sorted alphabetically by album

**Title:** Add Artist

**Actor:** User

**Requirements:** Req-2.3

**Main Scenario:**

1. System displays Add Artist form
2. User enters the name of the artist
3. System adds the artist to the application
4. System displays the Add Artist form

**Alternatives:**

- 3a. Name is empty or only spaces
- 3a1. System shows error message "Artist name cannot be empty!". Use case continues at step 1.

**Title:** Add Album

**Actor:** User

**Requirements:** Req-2.1

**Precondition:** An artist has already been created as per use case "Add Artist"

**Main Scenario:**

1. System displays Add Album form
2. User selects an artist, enters a name for the album, enters a genre for the album and enters the album's release date
3. System adds the album to the selected artist
4. System displays the My Music page

**Alternatives:**

- 3a. No artist is selected
- 3a1. System shows error message "Album must have an artist!". Use case continues at step 1.
- 3b. Album name is empty or only spaces
- 3b1. System shows error message "Album name cannot be empty!". Use case continues at step 1.
- 3c. Album genre is empty or only spaces
- 3c1. System shows error message "Genre name cannot be empty!". Use case continues at step 1.
- 3d. Release date is in the future
- 3d1. System shows error message "Release date cannot be in the future!". Use case continues at step 1.

**Title:** Add Song

**Actor:** User

**Requirements:** Req-2.2

**Precondition:** An album has already been created as per use case “Add Album”

**Main Scenario:**

1. System displays Add Song form
2. User selects an album, enters a name for the song, enters a duration for the song and enters the song's position in the album
3. System adds the song to the selected album
4. System displays the My Music page

**Alternatives:**

- 3a. No album is selected
- 3a1. System shows error message “Song must belong to an album!”. Use case continues at step 1.
- 3b. Song name is empty or only spaces
- 3b1. System shows error message “Song must have a name!”. Use case continues at step 1.
- 3c. Song duration is not a positive number
- 3c1. System shows error message “Song must have a duration!”. Use case continues at step 1.
- 3d. Song position is not a positive number
- 3d1. System shows error message “Song must have a position!”. Use case continues at step 1.

**Title:** Create Playlist

**Actor:** User

**Requirements:** Req-2.4

**Main Scenario:**

1. System displays create playlist form
2. User enters the name of the playlist
3. User selects which songs to put into playlist
4. System adds the playlist to the application
5. System displays the create playlist form

**Alternatives:**

- 3a. Name is empty or only spaces
- 3a1. System shows error message “Playlist name cannot be empty!”. Use case continues at step 1.

**Title:** Create Room

**Actor:** User

**Requirements:** Req-1.2

**Main Scenario:**

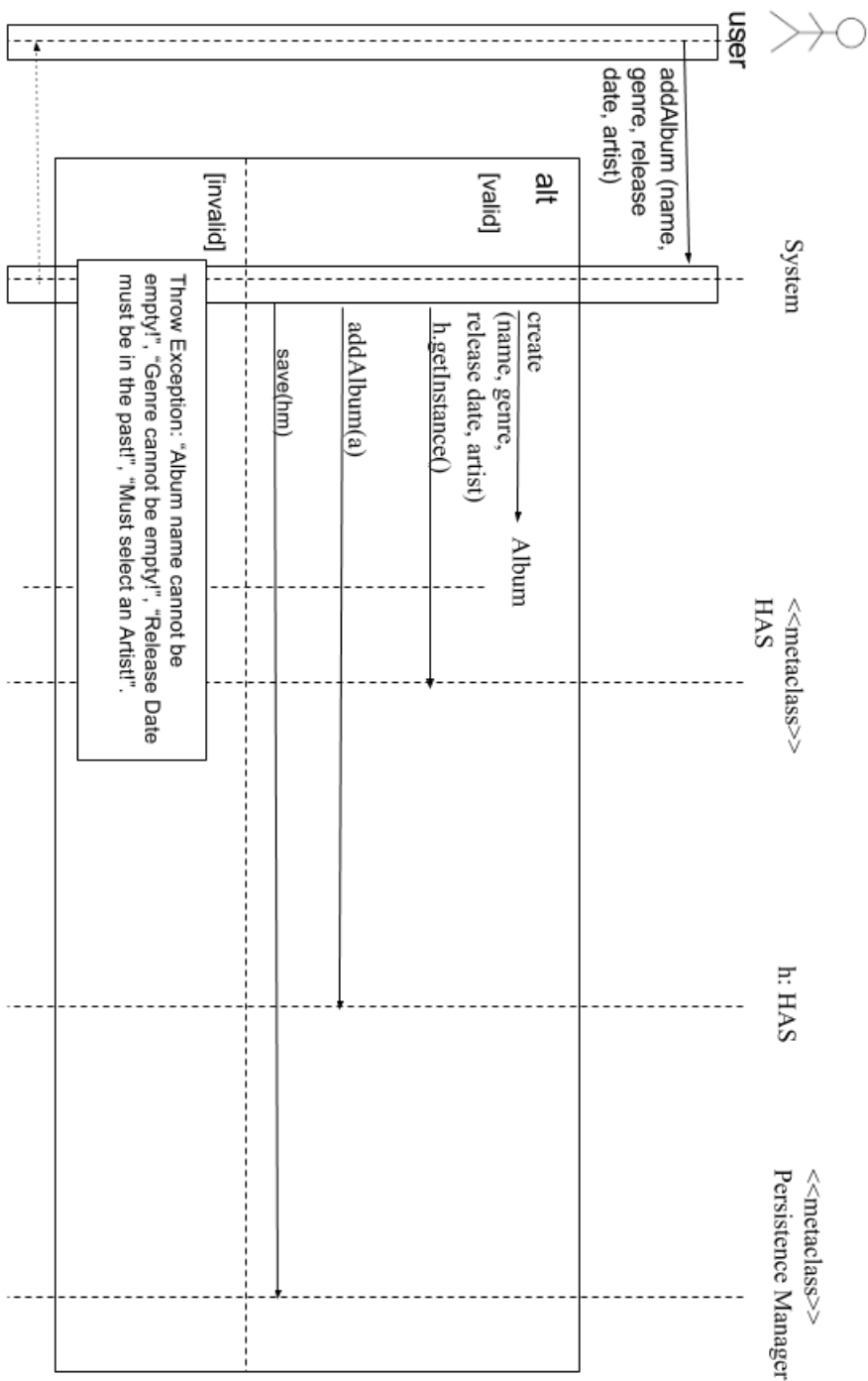
1. System displays Add Room form
2. User enters the name of the room
3. System adds the room to the application
4. System displays the Add Room form

**Alternatives:**

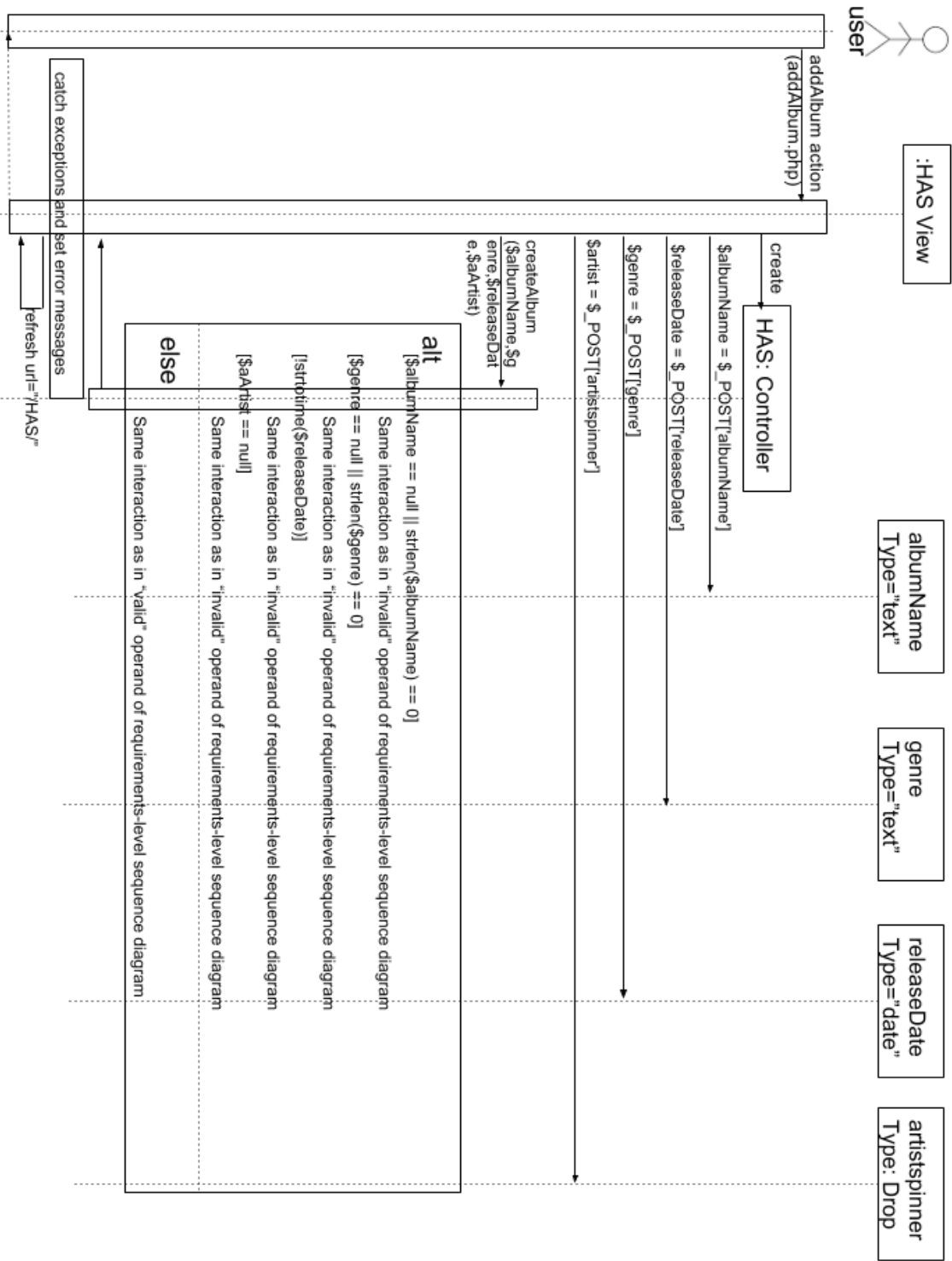
- 3a. Name is empty or only spaces
- 3a1. System shows error message “Room name cannot be empty!”. Use case continues at step 1.

When the user selects an item to play, they must then specify the room(s) to play the item in. A room can also be selected from the My Music home in order to control the volume in each room (Req-1.4) without having to select a new item to play.

Requirements Level Sequence Diagram

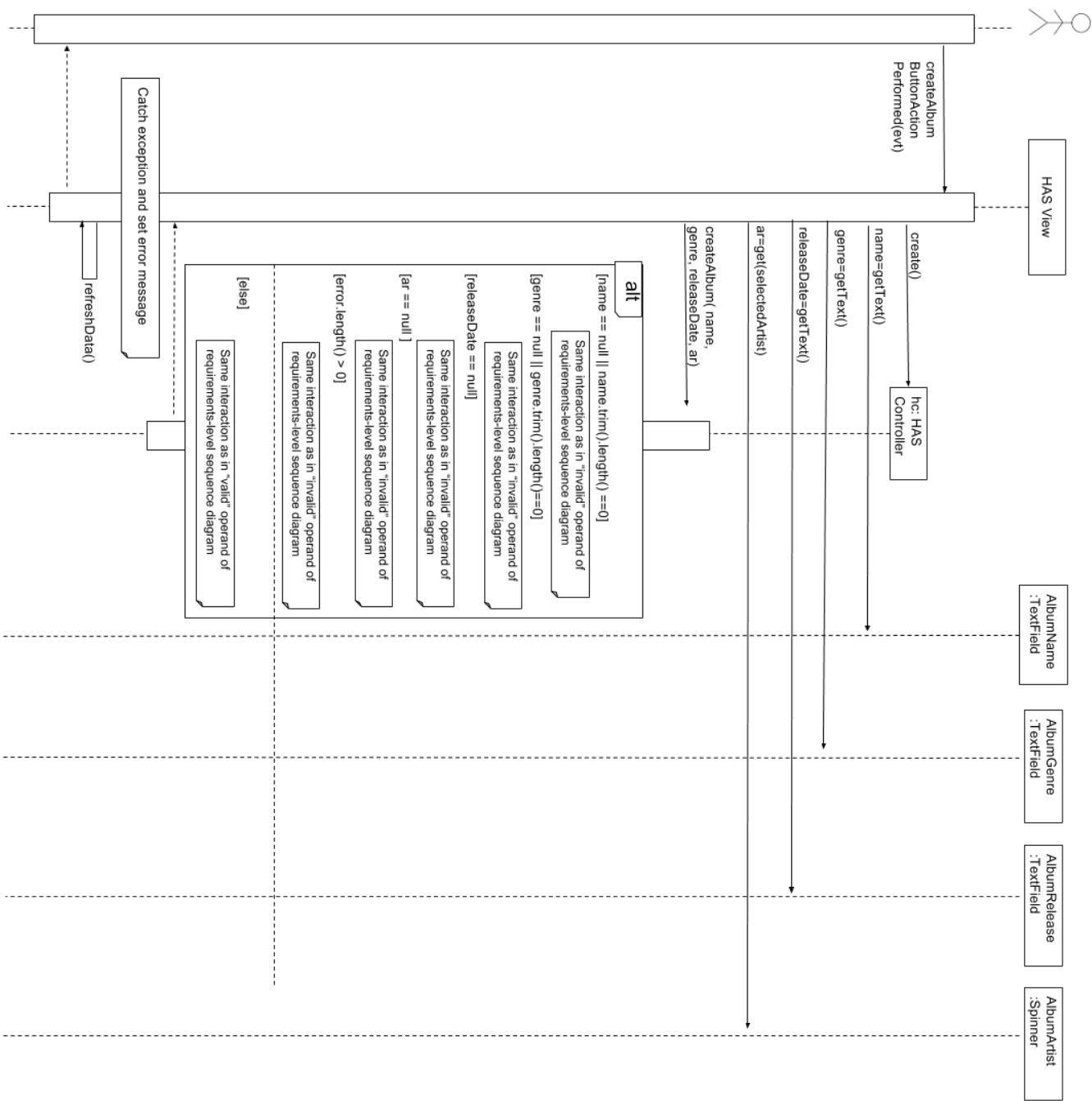


# Implementation-Level Sequence Diagram PHP





Implementation Level Sequence Diagram Desktop



Requirement-Level Sequence Diagram Android

