**System Sequence Diagrams:**

**Shared Alternative Path for all Use Cases:**

1. Server attempt to find and fix issues

1a. Server is offline

1. App informs the user that no server is detected

2. User restarts the serve

3. App reconnects to the server

1b. Server crash

1. Server auto restart

1a. Server fails to auto restart

1. Mobile app informs the user after 1 minute of no response from the server

2. User manually restarts server

1. Server requests information from the mobile application and resumes normal functionality

\*b At any time the mobile application fails:

1. Mobile app searches for issue

1a. Mobile app has no network connection

1. App attempts to connect to wifi to restore connection

1a. App fails to connect to wifi

1. The information is stored on the app and queue to send when connection is restored

2. The app informs the user that it has no network connection.

2a. Mobile application crashes

1. Mobile app sends information about the cause of the crash

2. App attempts to auto restart

2a. Fails to auto restart

1. User manually restarts the app

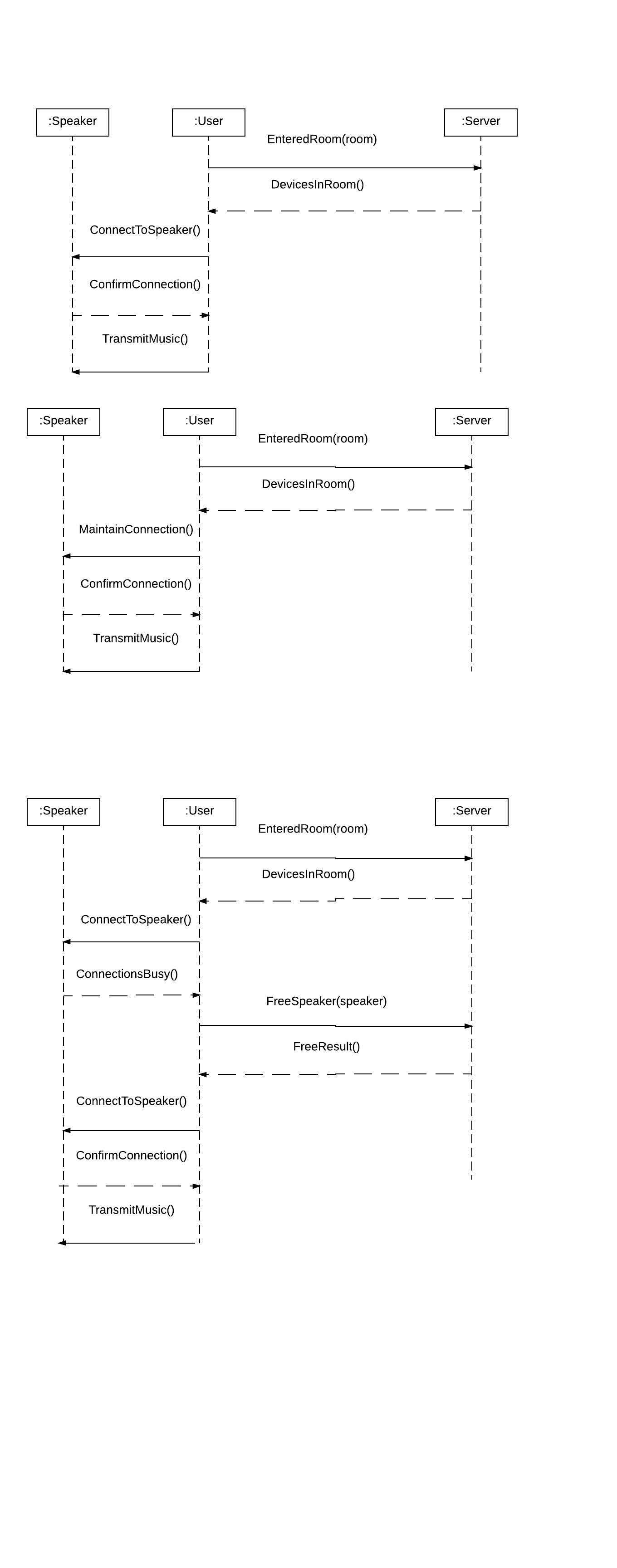
2. App sends information to the server

3. Mobile app sends its information to the server and reestablishes connections based on what the server sends back

**System Sequence Diagram:** Turning on a speaker when entering a room

**Main Success Scenario:**

1. The user walks into a new room
2. The mobile app sends a message to the server telling it what room it has entered.
3. The server receives the information and finds the speaker registered for that room. It sends that speaker’s information back to the mobile application.
4. The mobile app receives the speaker information and locates new speaker through bluetooth and connects to it.
5. New speaker is now playing music

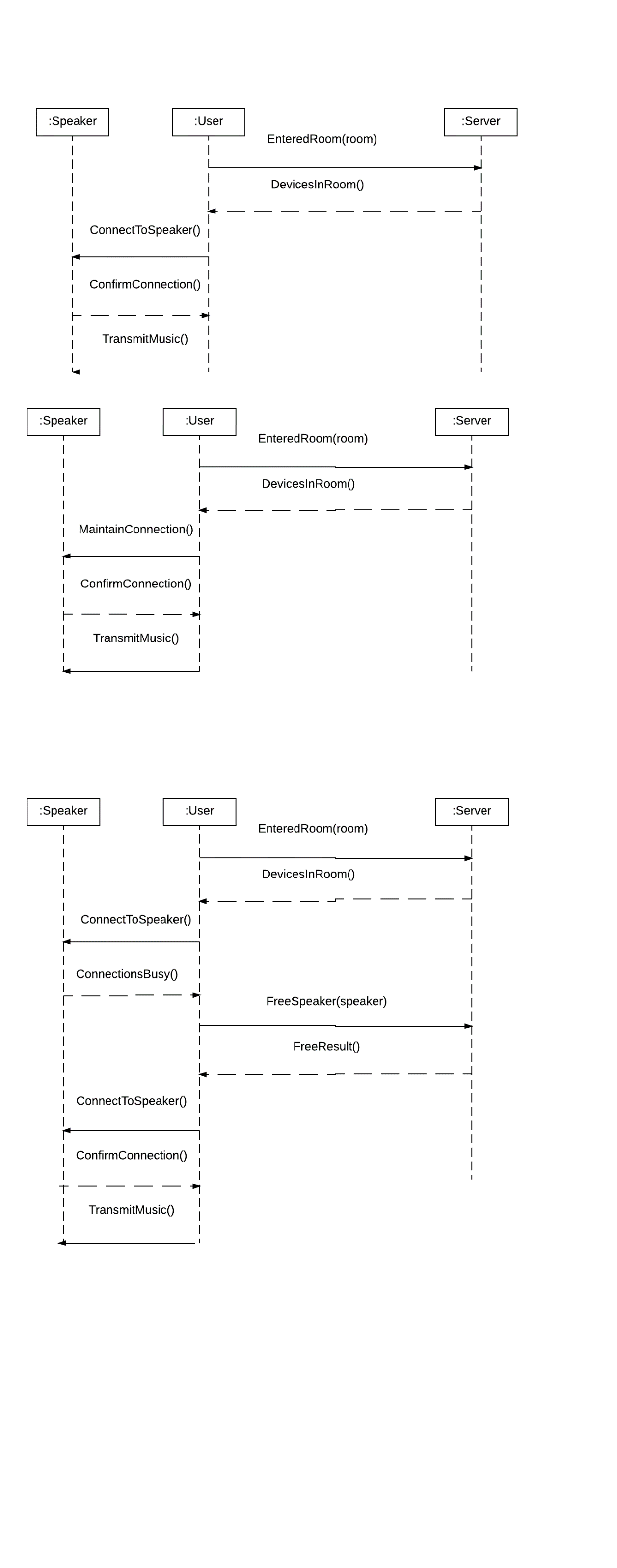
****

**Alternate Paths:**

3a. No speaker is found for that room

1. The system tells the mobile app that no new speaker is available

2. The mobile app remains connected to the old speaker



4a. Speaker is already connected to another bluetooth device

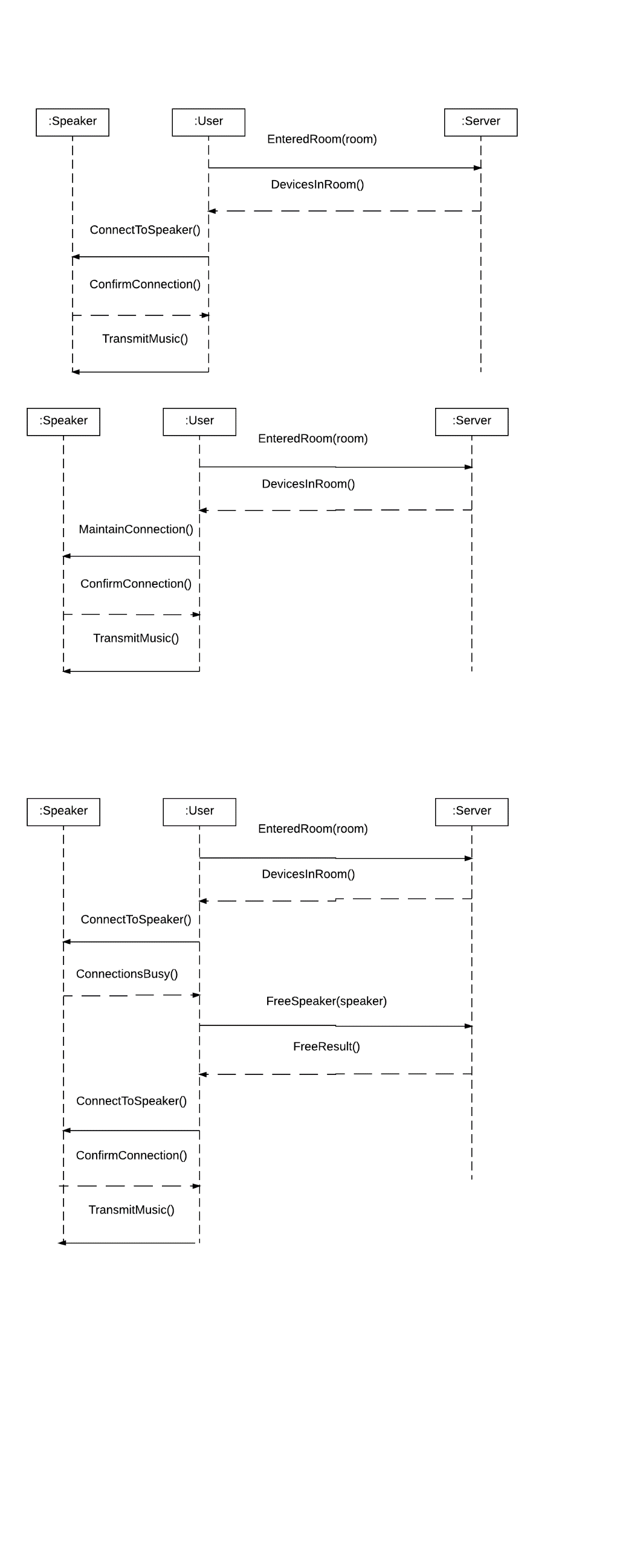
1. The mobile app informs user that the speaker is currently in use

2. User disconnects other device from the speaker

3. mobile app connects to speaker

3a. fails to connect to speaker

1. reconnects to old speaker



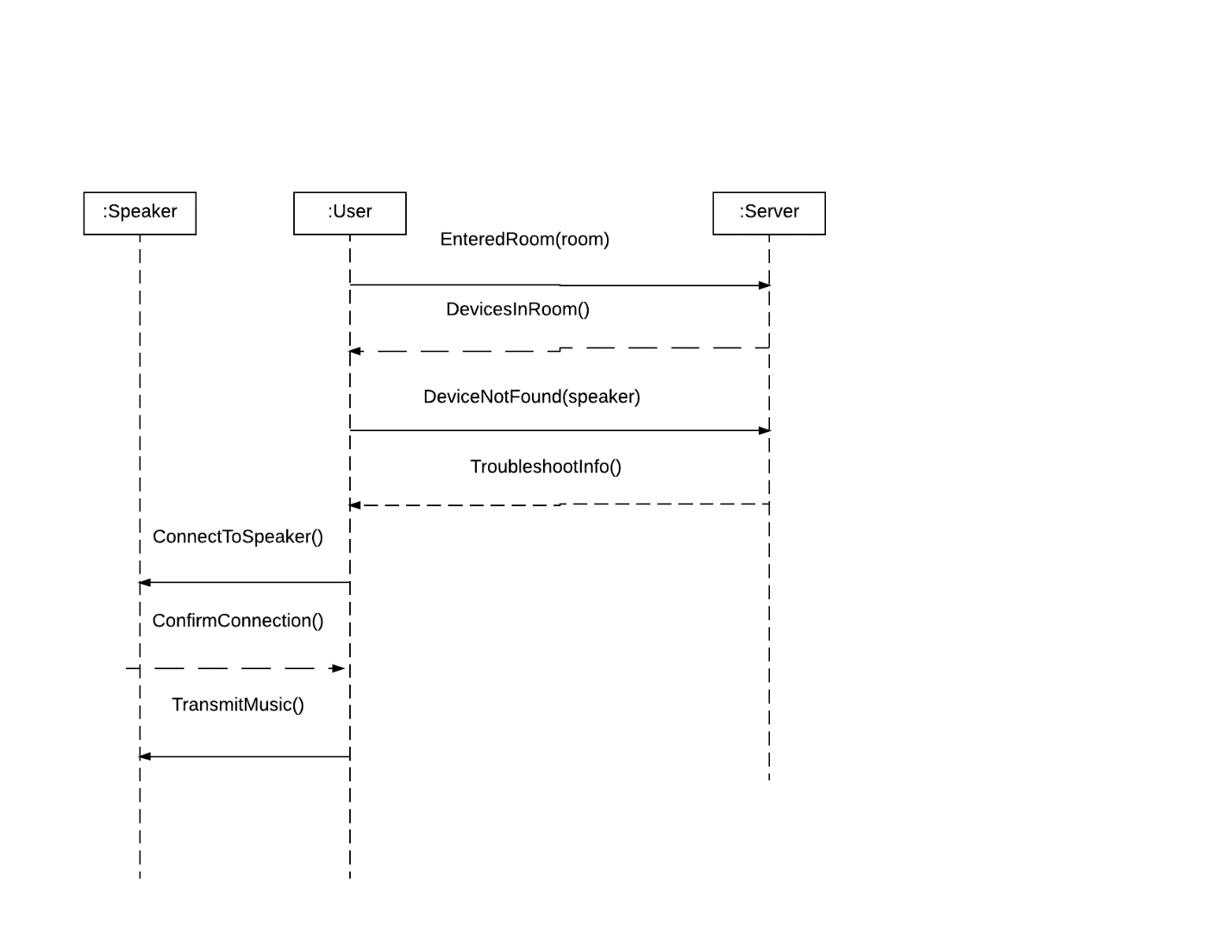
4b. Speaker is offline

1. The mobile app informs the user that the speaker is offline

2. User powers on the speaker

3. Mobile app connects to speaker

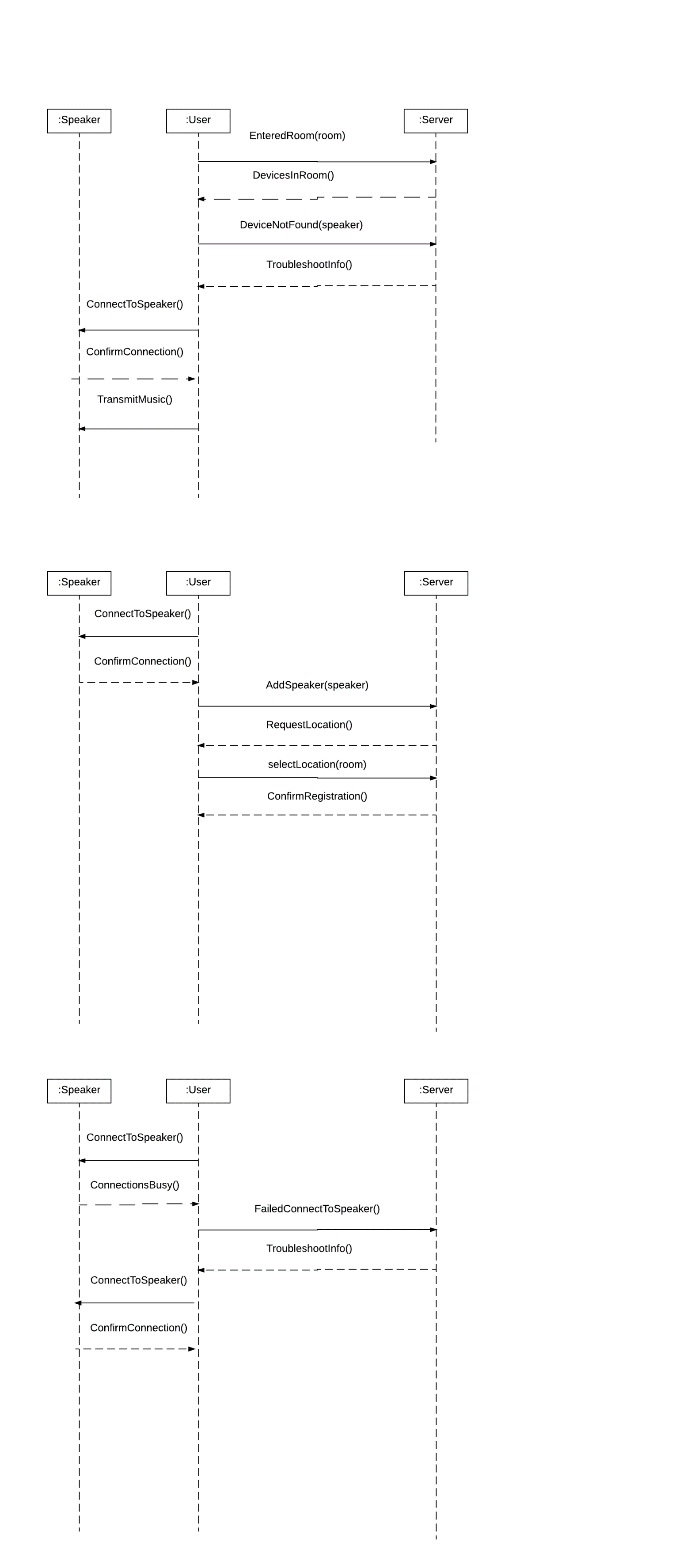
3a.fails to connect to speaker

1. reconnects to old speaker

**System Sequence Diagram:** Adding a new speaker to the system

**Main Success Scenario:**

1. The user uses the mobile application and navigates to the “add a device” menu
2. The mobile app displays the options of devices to add
3. The user selects the speaker option.
4. The mobile app instructs the user to connect to the speaker through bluetooth and then click register.
5. The user connects to the speaker and begins the register process.
6. The mobile app records the information for the speaker and asks the user to designate the room in which the speaker will be located.
7. The users selects the room manually or uses their mobile device location.
8. The mobile app sends the information to the server.
9. The server records the new speaker to its database.
10. The speaker is now registered.



**Alternative Paths:**

4a. The user cannot connect to the speaker

1. The user must ensure that the speaker is on, in pair mode and not connect to any other devices.

1a. The speaker is off

1. The user powers on the speaker and puts it in pair mode

1b. The speaker is not in pair mode

1. The user puts the speaker in pair mode

1c. The speaker is connected to another device

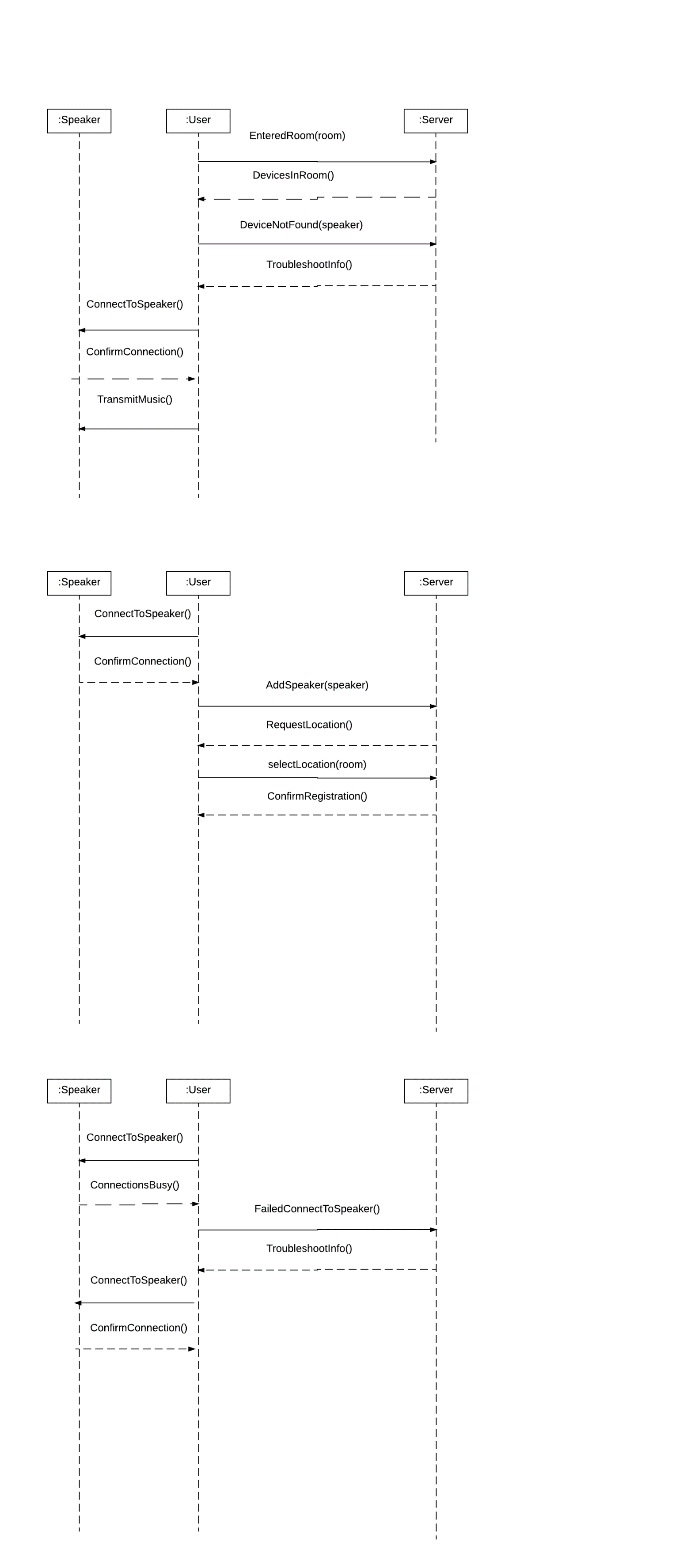
1. The user disconnects the other devices

2. The user attempts to connect to the speaker

2a. The connection fails

1. Attempt to connect again

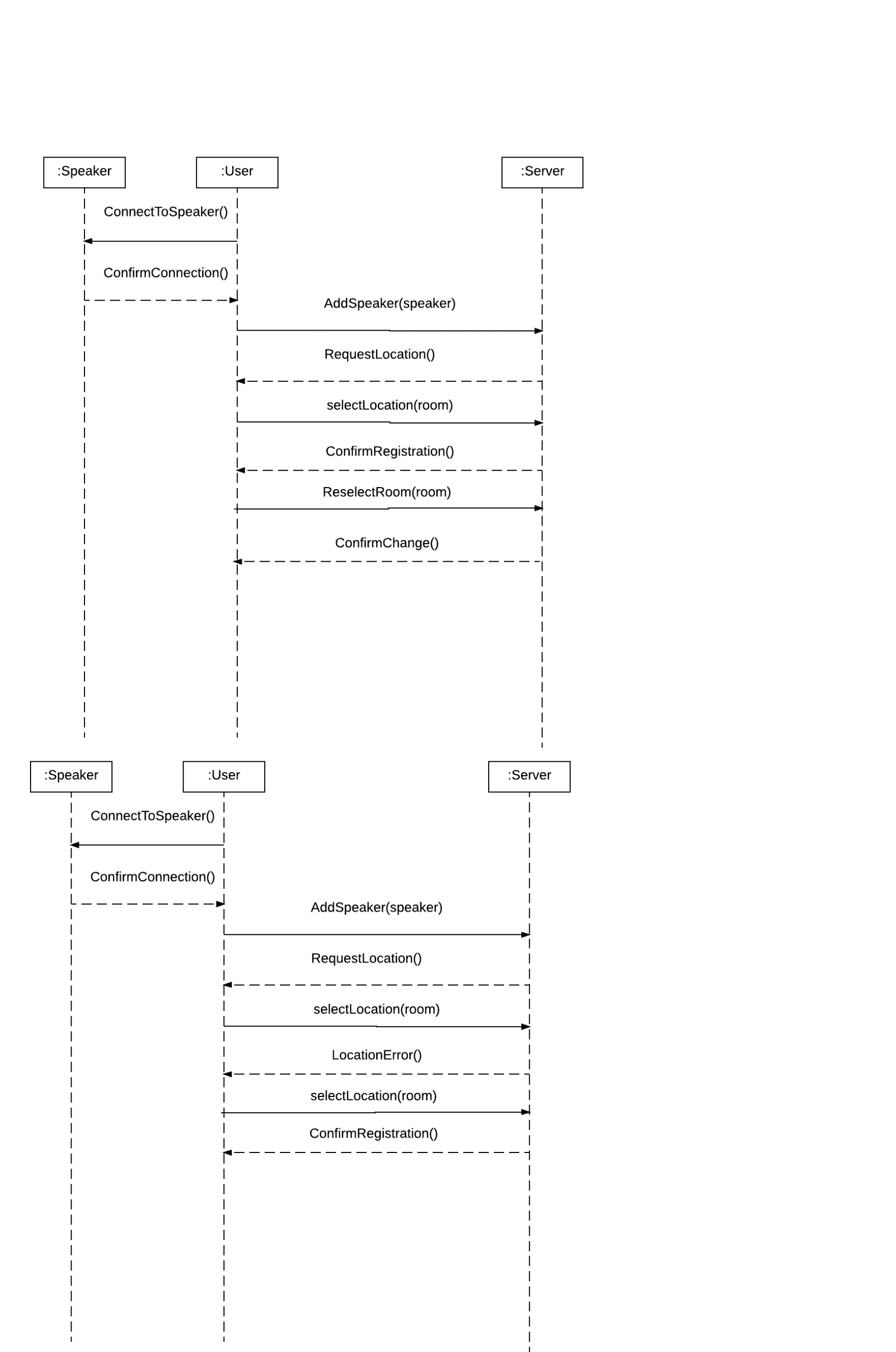
2. If it fails again, the speaker may be damaged.



6a. The use current location feature selects the wrong room

1. The user clicks the back arrow to try again

2. The user selects the correct room for the speaker



8a. The speaker has already been registered to another room

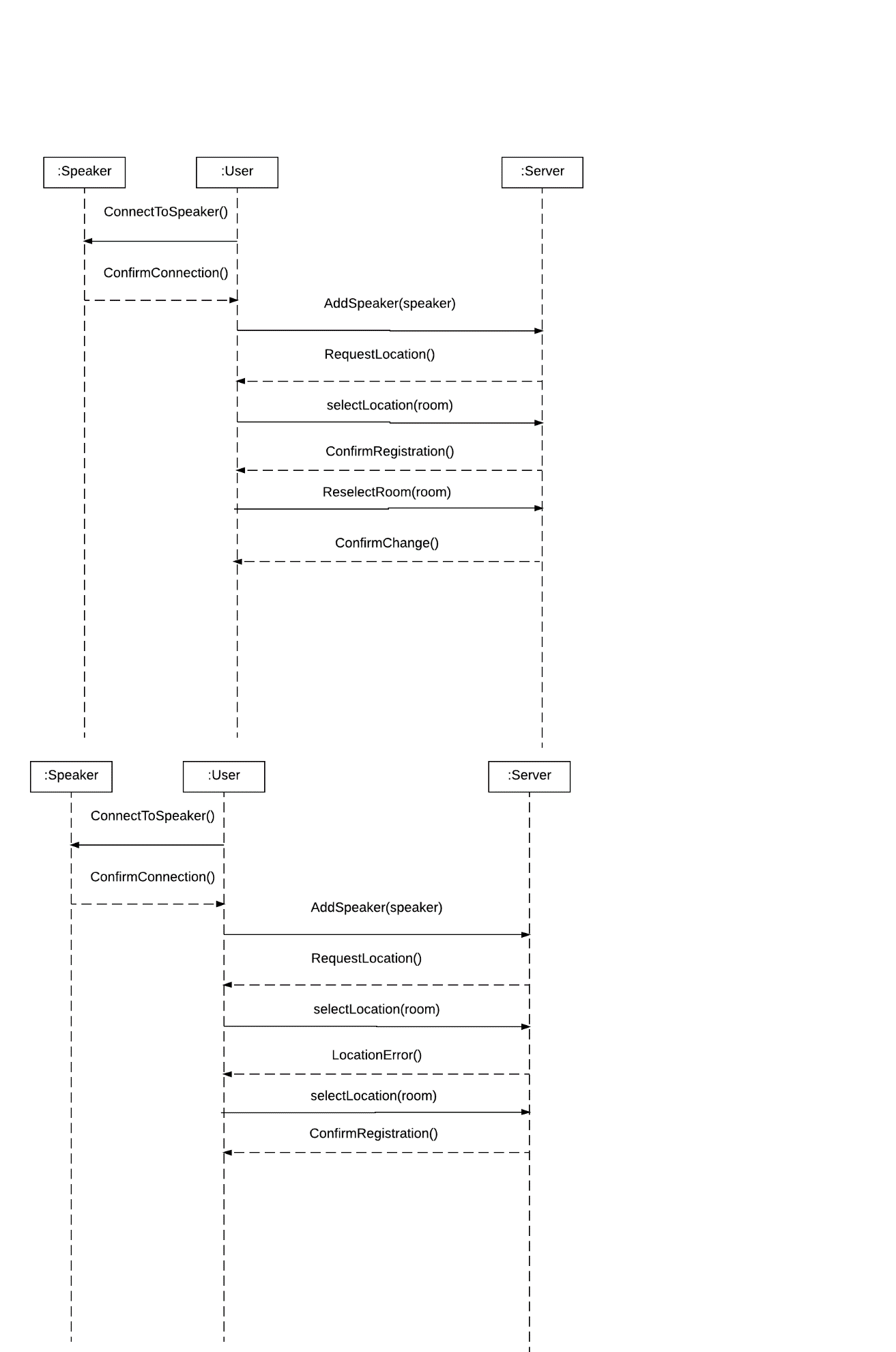
1. The server tells the mobile app that the device has been registered before and tells what room

2. The mobile app asks the user if they want to keep the old registered room or replace it

3. The user select their choice.

4. The mobile app informs the server of the decision

5. The server performs that action for the speaker.



**System Sequence Diagram:** Turning off lights when leaving a room

**Main Success Scenario:**

1. The user walks out of a room.
2. The mobile app sends a message to the server telling it what room it has just left.
3. The server receives the information and finds the light/lights registered for that room.
4. The server sends a signal to those lights to turn off.
5. The lights are now off in the old room.

**Alternative Paths:**

3a. No light is found for that room

1. The system makes a note of the missing light and sends it to the mobile app

2. The Mobile app stores this information in order to offer suggestions to the user at a later

time.

4a. Lights fail to turn off

1. Server tries to find error

1a. Light is not connected to the server

1. The server notes the communication error and sends it to the mobile app

2. The mobile app informs the user that the lights are not connected to the server

3. The user connects the light to the system

1b. Light is connected but fails to turn off

1. The server informs the mobile app of the error

2. The mobile app tells the user that the light isn’t functioning

2. The user disconnects and reconnects the light to the system

2. The light is recalibrated and attempts to turn off again.

2a. The light still won’t work with the system

1. The system tell the mobile app the lights may be faulty.

2. The mobile app tells the user that the lights may need to be replaced

3. User replaces the lights, disconnects the old lights and connects the new ones to the system

**System Sequence Diagram:** Change Password

**Main Success Scenario:**