# Use Case # [Calibrate room boundaries]

|  |  |
| --- | --- |
| GENERAL CHARACTERISTICS | |
| **Author** | Yibing Zhang |
| **Last Update:** | 9/25/2017 |
| **Scope** | Home Automation System |
| **Level** | User-goal |
| **Status** | Incomplete Conceptualization |
| **Primary Actor** | Mobile Application User |
| **Secondary Actors** | Server; Mobile Application |
| **Stakeholders and Interests** | Mobile App User: Wants to calibrate room boundaries to support multiple users have the music and lights on at the same time |
| **Preconditions** | User has Mobile application installed and registered.  One room is already occupied by someone else |
| **Success Post Condition** | Another user walking into a new room that is occupied by another user would not take control of the speaker and light (first come, first serve) |
| **Failed Post Condition** | Permission conflict occurs and the later guys takes control of everything |

|  |  |
| --- | --- |
| MAIN SUCCESS SCENARIO (or basic flow) | |
| **Step** | **Action -** description in words of each step in success scenario |
| 1 | The user walks into another room that is occupied by another user |
| 2 | The mobile app sends a message to the server telling it what room it has entered. |
| 3 | The server receives the information and check if the room is occupied by someone else. It sends that speaker’s information back to the mobile application. |
| 4 | The mobile app receives the speaker information and doesn’t connect to the speaker in this room. |
| 5 | The previous user still has access to the speaker in his room |

|  |  |
| --- | --- |
| EXTENSIONS or Alternate Flows | |
| **Step** | **Branching Action** |
| *n..m* | \*a At any time the server fails:   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 server  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 |
| 4. The user cannot connect to the speaker that is occupied by another user  1. The user must ensure that the speaker is not occupied  2. The user will have no connection with any speaker since he walks into a room that is occupied by anther user (he lost connection once he leaves his own room) |

|  |  |
| --- | --- |
| SPECIAL REQUIREMENTS | |
| **Req Num** | **Requirement** |
| *n* | 1. Speaker connections restricted to users 2. Server communications require authentication 3. Server communication allowed outside of the local network |

|  |  |
| --- | --- |
| TECHNOLOGY AND DATA VARIATIONS LIST | |
| **Var Num** | **Variation** |
| *n* | 7a. Communications are done over a wireless network so server would need internet and mobile device would need mobile data or be connected to wifi  7b. Keyboard is required to restart server as admin passwords would be needed |

***FREQUENCY OF OCCURRENCE***: Not often. It only occurs when the speaker is broken or the user gets a new speaker

|  |  |
| --- | --- |
| OTHER ISSUES | |
| **Issue Num** | **Issue** |
| *n* | 1. How will we deal with case once the user lost all connections when he walks into an occupied room? Should we let the mobile app keeps looking for the nearest speaker or we force it to stop? 2. How will the occupied room’s information be stored? |