|  |  |
| --- | --- |
| **Precondition**: The sensors two lasers are working and the sensor is on | |
| **Actor actions**  1. Venue-Hopper enters or exits a venue where sensor is located  3. Another Venue-Hopper enters or exits | **System Responses**  2. Sensor connects to the remote database and updates the appropriate in or out count  4. Repeat updating the database with venue’s population until sensor is turned off |
| **Postcondition**: Database contains result that was reported by sensor and Venue-Hopper is in or out of the venue | |

**Sensor – DetectPassingPerson**

**Mobile Application – CheckOutOwnedVenue**

|  |  |
| --- | --- |
| **Precondition:** User has account with Venue-Owner privileges | |
| **Actor actions**  1. Venue-Owner sends their login information to database  3. Request information for owned venue from the database  5. View venue information | **System Responses**  2. Validate login information  4. Send venue information corresponding to owner account |
| **Postconditions**: Database connection established and venue information is available for duration of user login | |

**Mobile Application –RegisterAccount**

|  |  |
| --- | --- |
| **Precondition** – User doesn’t have a preexisting account and has mobile app installed. | |
| **Actor actions**  1. User Indicates the need to register  3. User provides information. May request owner permissions    5. Confirmation message and user is brought to Map View (Figure 5) with Tutorial Overlay | **System Responses**  2. System asks for information such as name, e-mail, password, etc. to register.  4. System validates the information, checking for duplicates, then updates database with user information and privileges |
| **Postcondition**: The user has access to the application features depending on privilege level | |

**Remote Database – FindNearbyVenues**

|  |  |
| --- | --- |
| **Precondition**: Phone’s GPS is enabled for mobile app and user has a registered account.  Phone is within (TBD) radius of a venue with a sensor. | |
| **Actor actions**:  1. Logged in user enters the Map View and application requests locations in city  3. App finds and displays the venues that are within radius of user’s current GPS location.  User clicks on a nearby venue to view venue information    5. User views the information | **System Responses:**  2. Database verifies user account info and establishes session. Database returns locations in current user’s city.    4. Database returns information for clicked on venue |
| **Postcondition**: User has established a connection with database and can view nearby venues and information about them | |