

Functional Requirements

The system should allow users to:

1. User Authentication

- Register and log in using username and password.
- Only logged-in users can submit ratings. This will prevent review bombing/spamming.

2. View Campus Locations

- Browse a catalog of campus locations, including buildings and non-building spaces (e.g., open areas).
- Each location includes:
 - Name
 - Type (building/non-building)
 - Associated image
 - Average rating score

3. View Room Details

- Display details of rooms within buildings:
 - Room name
 - Room number
 - Type (e.g., Classroom, Study Room, Science Lab, Computer Lab)
 - Room size
 - Equipment available
 - Accessibility features (wheelchair, etc.)

4. Submit Ratings

- Users can rate rooms/locations using the following attributes:
 - Noise (scale)
 - Equipment (tags)
 - Cleanliness (scale)
 - Accessibility (tags)
 - Equipment Quality (categorical: Bad, Decent, Good)

- Wi-Fi strength
- Extra comments (free text)

5. Store & Retrieve Ratings

- Ratings are associated with both the user and the location.
- The system stores date-stamped records for each rating.
- Users can view all ratings they've submitted.

6. Manage Equipment & Accessibility Tags

- Admins or power users can manage curated tag sets for:
 - Equipment (e.g., projector, wheelchair ramp)
 - Accessibility (e.g., Braille signage, step-free access)

7. Multimedia Support

- Support images (BLOB format) for each location to enhance user experience.
-

Non-Functional Requirements

1. Performance

- The system should handle concurrent users efficiently (university environment w/ hundreds (or more?) on campus simultaneously).
- Responses for queries (e.g., searching locations or viewing room details) must return quickly so as to not alienate new or casual users.

2. Scalability

- Should be capable of handling additional campus locations, new room types, and more rating categories with minimal system change.

3. Security

- Passwords must be securely hashed and stored.
- Only authenticated users can post or edit ratings.
- Role-based access control for managing equipment/accessibility tags.

4. Usability

- The interface must be intuitive and mobile-responsive.
- Ratings should be easy to submit with guided dropdowns/sliders.
- Room types and sizes must be clearly defined to avoid user confusion.

5. Data Integrity

- Locations, Rooms, and Attributes are real places/things that exist in the real world.
- Input validations (e.g., rating values within allowed ranges).
- Use of constraints to enforce valid data (e.g., check constraint on rating quality fields).

6. Maintainability

- Clear data model using structured relationships (as defined in the ER diagram).
- Codebase should follow modular design to facilitate updates and testing.

Simulated Stakeholder Interview Insights

- **Campus Admin:** Wants to track student feedback on study spaces for future renovations.
- **Students:** Need a way to find quiet or accessible rooms based on their preferences.
- **IT Team:** Requests easy tagging and rating functionality to avoid database overload.