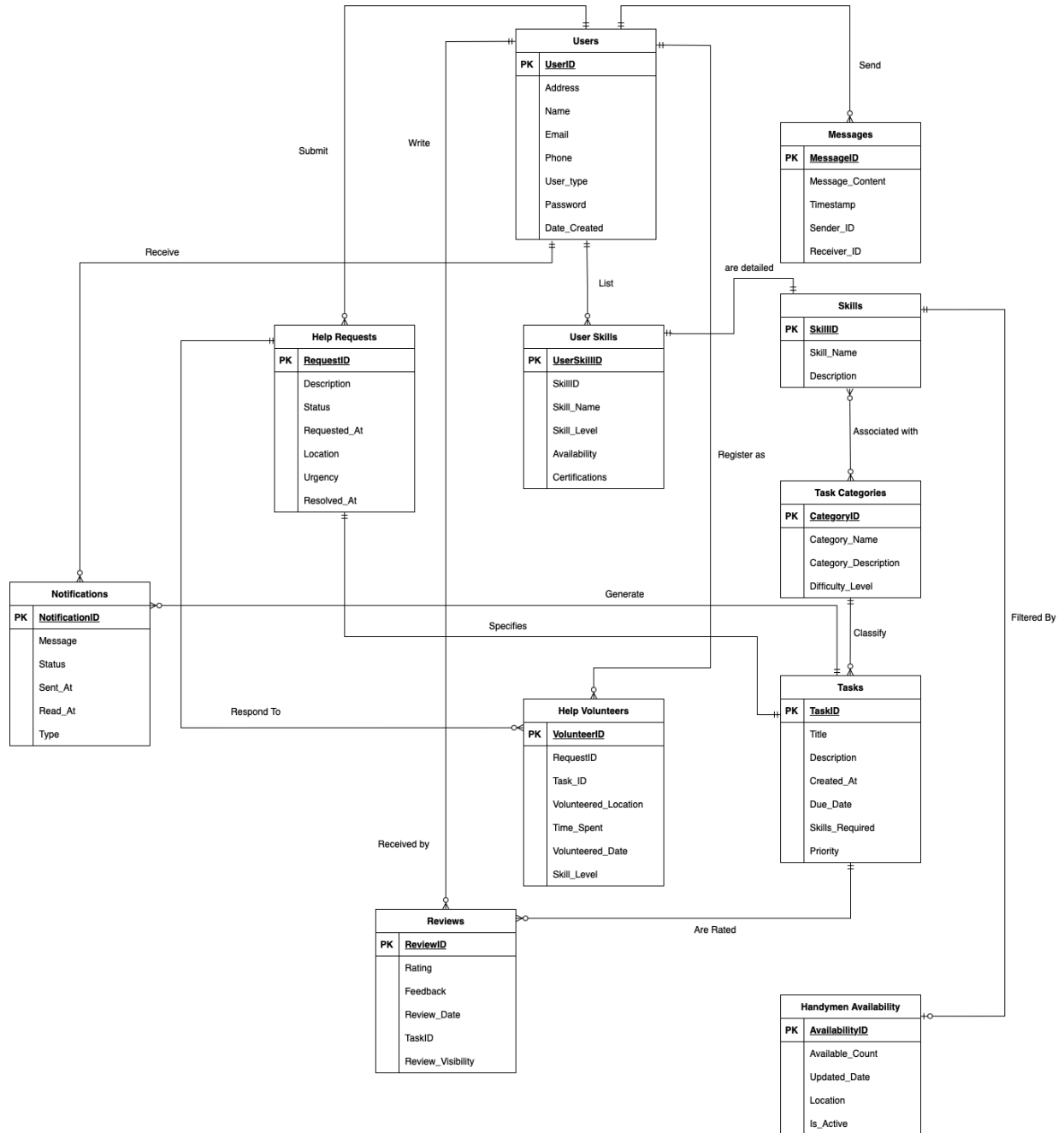


## P2. Handyhood

### ERD Diagram



## Entities and Attributes

### 1. User

- Attributes: UserID (PK), Name, Email, Password, Phone, Address
- Purpose: Manages user profiles, login credentials, and contact details for community members seeking or offering help.

### 2. Handyman (Specialized User)

- Attributes: Inherits User attributes + Skillset, AvailabilityStatus, Rating
- Purpose: Tracks skilled volunteers, their availability, and expertise (linked to the *Skill* entity).

### 3. Task

- Attributes: TaskID (PK), Title, Description, Category, UrgencyLevel, Status
- Purpose: Stores predefined tasks/issues requiring skilled assistance (e.g., plumbing, electrical work).

### 4. Skill

- Attributes: SkillID (PK), SkillName, Description
- Purpose: Maintains a catalog of skills (e.g., carpentry, painting) to match tasks with qualified handymen.

### 5. HelpRequest

- Attributes: RequestID (PK), CreationDate, CompletionDate, Status
- Purpose: Records issues raised by users and tracks their lifecycle from submission to resolution.

### 6. Review

- Attributes: ReviewID (PK), Rating, Comment, Date
- Purpose: Stores feedback on handymen's performance for transparency and quality control.

### 7. Notification

- Attributes: NotificationID (PK), Message, Timestamp, Status
- Purpose: Alerts handymen about new tasks and updates request status (e.g., "assigned," "completed").

### 8. SkillAvailability (Associative Entity)

- Attributes: HandymanID (FK), SkillID (FK), AvailableCount
- Purpose: Tracks the number of handymen available per skill for real-time resource allocation.

## Relationships and Cardinalities

1. User submits HelpRequest
  - Type: 1:N
  - A user can submit multiple help requests, but each request is tied to one user<sup>1</sup>.
2. HelpRequest requires Skill
  - Type: M:N (via *Task* entity)
  - Tasks are categorized by skills, allowing multiple skills to be linked to a request<sup>1</sup>.
3. Handyman volunteers for HelpRequest
  - Type: M:N (resolved via *Volunteer* associative entity)
  - Attributes: VolunteerID, AcceptanceStatus
  - Enables multiple handymen to volunteer for a request, with one ultimately assigned<sup>1</sup>.
4. Handyman possesses Skill
  - Type: M:N (via *SkillAvailability*)
  - Tracks which handymen are qualified for specific skills and their availability<sup>1</sup>.
5. Notification triggers Task update
  - Type: 1:1
  - Notifications directly update task status (e.g., “pending” → “in progress”)<sup>1</sup>.
6. Review associates HelpRequest and Handyman
  - Type: 1:1 per task
  - Ensures each completed task has one review linked to both the handyman and request<sup>1</sup>.

## System Workflow Alignment

- User Management: User and Handyman entities store profiles and skills.
- Task/HelpRequest Management: Task defines requirements; HelpRequest tracks issue resolution.
- Skills Availability: SkillAvailability dynamically updates handyman counts per skill.
- Notifications: Automated alerts ensure real-time task status updates.
- Reviews: Feedback loop improves service quality and accountability.

This design ensures efficient matching of community needs with available skilled volunteers while maintaining transparency through notifications and reviews.