

## Acceptance Criteria

### I. ERD

1. The ERD should represent entities for user, task, group, achievements, rewards, activity\_log, and notifications with their appropriate relationships.
2. User entity should have:
  - a. Email field (unique, required)
  - b. Password field (encrypted, required)
  - c. Username field (unique, required)
  - d. Role field (member/admin/system\_admin)
3. Task entity should have:
  - a. Title field (required, string)
  - b. Description field (text, optional)
  - c. Points value field (integer, required)
  - d. Medal tier field (enum: bronze/silver/gold/platinum)
  - e. Status field (enum: available/in\_progress/completed/pending\_approval)
  - f. Deadline field (datetime, optional)
  - g. Created at timestamp
  - h. Updated at timestamp
  - i. Completion timestamp (nullable)
  - j. Creator ID (foreign key to User)
  - k. Assignee ID (foreign key to User, nullable)
  - l. Group ID (foreign key to Group)
  - m. Recurrence settings (JSON or enum: none/daily/weekly/custom)
  - n. Attachments field (JSON array or separate table)
4. Group entity should have:
  - a. Group name field (required, string)
  - b. Description field (text, optional)
  - c. Created at timestamp
  - d. Updated at timestamp
  - e. Creator ID (foreign key to User)
  - f. Group settings (JSON: member permissions, point rules, etc.)
  - g. Invite code field (unique string for joining)
  - h. Maximum members field (integer, optional)
5. Achievements entity should have:
  - a. Description field (text)
  - b. Badge type field (enum: milestone/completion/streak/special)
  - c. Requirements field (JSON: conditions to earn badge)
  - d. Icon/image field (string URL or file path)
  - e. Points threshold field (integer, for milestone badges)

6. Rewards entity should have:
  - a. Reward name field (required, string)
  - b. Description field (text)
  - c. Point cost field (integer, required)
  - d. Quantity limit field (integer, optional)
  - e. Group ID (foreign key to Group)
  - f. Created by ID (foreign key to User - admin)
  - g. Active status field (boolean)
7. Activity log entity should have:
  - a. Event type field (enum: task\_created/task\_completed/points\_awarded/etc.)
  - b. Description field (text)
  - c. Actor ID (foreign key to User)
  - d. Target ID (polymorphic: Task/User/Reward ID)
  - e. Group ID (foreign key to Group)
8. Notifications entity should have:
  - a. Title field (required, string)
  - b. Message field (text, required)
  - c. Type field (enum: task\_reminder/completion/claim/reward/etc.)
  - d. Recipient ID (foreign key to User)
  - e. Read status field (boolean, default false)
  - f. Scheduled for field (datetime, for future notifications)
  - g. Sent at timestamp (nullable)
9. Junction/Relationship Tables
  - a. User-Group Membership
    - i. User ID (foreign key)
    - ii. Group ID (foreign key)
    - iii. Role in group (enum: member/admin)
  - b. User-Achievement
    - i. User ID (foreign key)
    - ii. Achievement ID (foreign key)
    - iii. Earned at timestamp
    - iv. Group ID (foreign key to Group, if group-specific)
  - c. Reward Redemptions
    - i. User ID (foreign key)
    - ii. Reward ID (foreign key)
    - iii. Redeemed at timestamp
  - d. Key Relationships
    - i. One-to-Many: User → Tasks (creator)
    - ii. One-to-Many: User → Groups (creator)
    - iii. One-to-Many: Group → Tasks
    - iv. One-to-Many: Group → Rewards
    - v. Many-to-Many: User ↔ Groups (membership)

- vi. Many-to-Many: User ↔ Achievements (earned badges)
- vii. One-to-Many: User → Activity Log (actor)
- viii. One-to-Many: User → Notifications (recipient)
- ix. One-to-Many: Task → Activity Log (target)
- e. Data Integrity Requirements:
  - i. All foreign key constraints should be properly defined
  - ii. Cascade delete rules should be specified (e.g., deleting a group should handle associated tasks)
  - iii. Unique constraints on email, username, group invite codes
  - iv. Check constraints on points (non-negative), status enums
  - v. Audit fields (created\_at, updated\_at) should be present on all primary entities
  - vi. Soft delete capability should be considered for critical entities (users, groups, tasks)

## II. Prototype

### 1. Login / Signup Page:

- a. Login Security
  - i. Login should use secure connection
  - ii. Passwords should be hidden when typing
  - iii. Show error message for wrong email/password
- b. Credential Validation
  - i. Check if email format is correct
  - ii. Both email and password are required
  - iii. Show clear error messages for invalid inputs
- c. Non-existing User Handling
  - i. If email doesn't exist, show "Sign up instead?" message
  - ii. Redirect to signup form easily
- d. Forgot Password
  - i. "Forgot password?" link should work
  - ii. User enters email and gets reset link
  - iii. Reset link should allow creating new password
- e. External Account Login
  - i. Show Google/Microsoft/Facebook login buttons
  - ii. Handle login with external accounts
  - iii. Create account automatically for first-time external users
- f. Email Verification
  - i. Send verification email after signup
  - ii. User must verify email before logging in
  - iii. Allow resending verification email
- g. Loading States

- i. Show spinner on buttons when processing
    - ii. Disable form during submission
    - iii. Show loading feedback to user
  - h. Password Requirements
    - i. No spaces allowed
    - ii. Minimum 8 characters
    - iii. Must have uppercase, lowercase, number, and special character
    - iv. Show which requirements are met/not met
    - v. Confirm password must match
- 2. Tasks Page
  - a. All Tasks Summary
    - i. Show all user's tasks when page loads
    - ii. Display task title, points, deadline, and status
    - iii. Show total task count
  - b. In Progress Tasks
    - i. Filter to show only tasks user is working on
    - ii. Show deadline and priority
    - iii. Allow completing tasks from this view
  - c. Completed Tasks
    - i. Show tasks user has finished
    - ii. Display completion date and points earned
    - iii. Show any medals/badges earned
  - d. Task Details
    - i. Hover over task shows quick preview
    - ii. Click task opens detailed view
    - iii. Show full description, attachments, and history
    - iv. Provide action buttons (claim, complete, edit)
- 3. Groups Page
  - a. Groups Summary
    - i. Display user's groups in cards
    - ii. Show group name, member count, recent activity
    - iii. Show user's role (member/admin)
  - b. Group Members
    - i. Click members icon to see member list
    - ii. Show member names and roles
    - iii. Display member points/rankings
  - c. View All Groups
    - i. "View All" button shows complete list
    - ii. Add search functionality

- iii. Show available groups to join
  - d. Group Navigation
    - i. Click group card goes to group page
    - ii. Group page shows group tasks and settings
    - iii. Provide admin controls if user is admin
- 4. Shop Page
  - a. Available Rewards
    - i. Show all rewards when page loads
    - ii. Display in grid layout
    - iii. Add search and filter options
  - b. Reward Details
    - i. Show reward name, description, cost
    - ii. Display reward type and availability
    - iii. Show quantity limits if any
  - c. Reward Redemption
    - i. Check if user has enough points
    - ii. Show confirmation before redeeming
    - iii. Update point balance after redemption
  - d. User Availability
    - i. Highlight rewards user can afford
    - ii. Gray out expensive rewards
    - iii. Show how many more points needed
- 5. Profile Page
  - a. Points and Medals
    - i. Display current points prominently
    - ii. Show earned medals collection
    - iii. Show points breakdown
  - b. Leaderboard Rank
    - i. Display current rank in groups
    - ii. Show rank position (e.g., "5th out of 20")
  - c. Badges
    - i. Show all earned badges
    - ii. Display badge descriptions
    - iii. Show progress toward next badges
  - d. Edit Profile
    - i. Allow editing username and contact info
    - ii. Upload profile picture
    - iii. Change notification settings
    - iv. Require password for important changes
- 6. Activity Page

- a. Task History
    - i. Show list of user's task activities
    - ii. Display dates and points earned
    - iii. Filter by date range
  - b. Shop Transactions
    - i. Show reward redemption history
    - ii. Display transaction status
    - iii. Show remaining point balance
  - c. Group Updates
    - i. Show group-related activities
    - ii. Display new members and announcements
    - iii. Link to relevant groups
7. Support Page
- a. FAQs
    - i. Display common questions and answers
    - ii. Organize by categories
    - iii. Add search functionality
    - iv. Make answers expandable/collapsible
  - b. Contact Us
    - i. Provide contact form (name, email, message)
    - ii. Show expected response time
    - iii. Send confirmation after submission
8. Logout
- a. Logout Process
    - i. Clear logout button in menu
    - ii. End user session completely
    - iii. Redirect to login page
    - iv. Show "logged out successfully" message
9. General Requirements
- a. Navigation: Consistent menu across all pages
  - b. Responsive: Works on mobile and desktop
  - c. Loading: Show loading states for all actions
  - d. Errors: Display clear error messages
  - e. Accessibility: Easy to use for all users