

Business Objectives:

1. Market Expansion and Revenue Growth

- Capture market share in the gamified productivity app sector by targeting teams, organizations, and communities seeking engaging task management solutions
- Generate revenue through premium features, in-app purchases for reward shop customization, and enterprise licensing for organizations
- Scale user base through viral growth mechanisms inherent in the gamification and social features

2. Enterprise Solution Independence

- Provide organizations with a comprehensive, self-contained task management platform that eliminates dependency on multiple external HR and productivity tools
- Reduce operational costs for businesses by consolidating task assignment, performance tracking, team engagement, and reward management into a single application
- Offer customizable enterprise features that adapt to specific organizational workflows and hierarchies

3. User Engagement and Retention Maximization

- Achieve high user retention rates through sustained engagement via gamification elements (points, badges, leaderboards)
- Create habitual usage patterns through AI-driven personalized task suggestions and optimal timing recommendations
- Foster community building within user groups to increase platform stickiness

4. Data-Driven Product Evolution

- Leverage user behavior analytics and AI insights to continuously improve task suggestion accuracy and engagement features
- Create valuable data assets around productivity patterns and team dynamics for potential B2B insights services
- Develop predictive capabilities for task completion and team performance optimization

User requirements:

1. Landing & Authentication (Group Leader & Member)

- The **user** shall sign up with email or social login (Google, Facebook, Apple) in order to quickly create an account and start using the app without issues.
- The **user** shall log in with their registered account in order to access their personalized task lists, points, and progress.

- The **user** shall switch easily between sign-up and login screens in order to ensure smooth navigation and reduce user frustration.

2. Task Management (Task Tab)

Member

- The **member** shall view today's date and navigate the calendar in order to see past, current, and upcoming tasks.
- The **member** shall select a date to view all tasks with point values in order to understand workload and motivation for task completion.
- The **member** shall filter tasks by status (in-progress, completed) in order to track progress and stay organized.
- The **member** shall claim and complete tasks to earn points in order to stay motivated and contribute fairly to group responsibilities.

Group Leader

- The group leader shall assign or rotate tasks among members in order to ensure fairness and prevent uneven workload distribution.

3. Activity Tracking (Activity Tab)

Member

- The **member** shall view team members' accomplishments and claimed rewards in order to encourage transparency, accountability, and positive competition.

Group Leader

- The **group leader** shall monitor member progress in order to oversee group performance and adjust responsibilities if needed.

4. Rewards & Motivation (Shop)

Member

- The **member** shall redeem earned points in the in-app shop in order to exchange productivity for tangible and fun rewards.
- The **member** shall choose higher-value rewards for bigger accomplishments in order to gain a sense of achievement and motivation for sustained participation.

Group Leader

- The **group leader** shall set custom rewards and assign point values in order to motivate members based on context-specific incentives.

5. Group Collaboration (Groups Menu)

Member

- The **member** shall create or join groups in order to organize specific responsibilities across different aspects of life (family, dorm, teams).

Group Leader

- The **group leader** shall manage group membership (approving, adding, or removing members) in order to maintain order and accountability.

6. Profile & Progress (Profile Page)

Member

- The **member** shall view total points, completed tasks, badges, medals, and leaderboard rank in order to track progress and stay engaged through gamified feedback.
- The **member** shall compare achievements with peers on the leaderboard in order to encourage friendly competition and higher participation.

Group Leader

- The **group leader** shall review member performance via points, completed tasks, and rankings in order to identify strong contributors and assist underperforming members.

7. AI-Powered Task Suggestions

Member

- The **member** shall receive AI-assisted task suggestions based on past entries, repetitions, and common keywords in order to save time and optimize consistency in daily routines.
- The **member** shall accept or dismiss AI-suggested tasks in order to maintain control while benefiting from smart automation.

Group Leader

- The **group leader** shall review AI-suggested group tasks in order to incorporate relevant recommendations into team responsibilities.

Functional Requirements:

1. Creating New Users (Self-Registration)

Purpose: Allow end users (family members, groupmates) to create their own TaskTrek account.

- **Actors:** New User (customer/member)
- **Preconditions:** Device connected to internet; app installed/open; email not already registered.
- **Inputs (required fields):**
 - Email (string, must be valid)
 - Password (string, min. 8 chars, complexity rules)
 - Confirmed Password
 - Username
 - Contact Number (optional or required per admin settings)
 - (Optional) Profile picture upload
- **Process:**
 - User opens Sign Up screen and fills form.
 - System validates fields (email format, password match & strength).

- System sends verification email (if email verification enabled).
- On successful verification, account created and user is redirected to onboarding or group-join flow.
- **Outputs:**
 - New user account record in DB
 - Verification email (if enabled)
 - Welcome/Onboarding screen
- **Postconditions:** User can log in. Profile record created.
- **Business rules / validations:**
 - Email must be unique.
 - Passwords must match and meet complexity.
 - Contact number format validated (country code optional).
- **Error handling:** Show clear messages for duplicate email, weak password, network error.
- **Notes:** Option to sign up via OAuth (Google/Facebook) may be implemented as alternate flow.

2. Creating Users (Admin Creation)

Purpose: Allow group admins (or system administrators, in some deployments) to create user accounts or add members.

- **Actors:** Group Admin, System Admin
- **Preconditions:** Admin is authenticated and authorized.
- **Inputs:**
 - Email
 - Temporary Password (or send invite link)
 - Username
 - Contact number (optional)
 - Role (member / admin)
 - Optional starting points/badges

- **Process:** Admin fills "Add Member" form → system sends invite email with activation / set-password link.
- **Outputs:** Pending user invite record; notification to invitee.
- **Business rules:** Admin cannot create another admin unless permitted.
- **Error handling:** Duplicate email prevented; show reason.

3. Login for Users

Purpose: Authenticate users and route them to their dashboard.

- **Actors:** Registered User
- **Preconditions:** User has an active account.
- **Inputs:**
 1. Email
 2. Password
 3. (Optional) Two-factor authentication code
- **Process:**
 1. User submits credentials.
 2. System validates credentials and 2FA if enabled.
 3. Determine user role(s) and redirect to appropriate landing page (Tasks view).
- **Outputs:** Auth token/session; user dashboard.
- **Postconditions:** User session established; last-login timestamp updated.
- **Business rules:** Lock account after N failed attempts; require password reset flow for forgot password.
- **Error handling:** Invalid credentials message; locked account notice.

4. Editing User Information (Profile Management)

Purpose: Allow users to update profile and contact information.

- **Actors:** Authenticated User
- **Preconditions:** User is logged in.

- **Editable fields:**
 - Profile name/display name
 - Profile picture
 - Email (requires re-verification)
 - Contact number
 - Notification preferences (email/push)
 - Privacy settings (who sees profile)
- **Process:** User updates fields → system validates → saves changes → returns success.
- **Outputs:** Updated user record; confirmation message; if email changed, verification flow started.
- **Business rules:** Email change requires verification before applied; changing role only via admin.
- **Error handling:** Invalid file upload, unsupported image type, failed verification.

5. Creating Groups

Purpose: Allow users (usually admins) to create organizational/group contexts (family, dorm, team).

- **Actors:** Authenticated User (with permission)
- **Preconditions:** User logged in.
- **Inputs:**
 - Group name
 - Group description
 - Group avatar/icon
 - Default point settings (optional)
 - Admin(s) selection
 - Invite method (email invite, code, link)
- **Process:** Create group record → initialize settings → send invites (if provided).
- **Outputs:** New group with unique ID; invite links/codes.

- **Business rules:** Group name must be unique within user's account (optional).
- **Error handling:** Invalid invite email, max groups limit per user if any.

6. Managing Group Membership (Invite / Remove / Promote)

Purpose: Add, remove, or change role of group members.

- **Actors:** Group Admin
- **Preconditions:** Admin is authenticated.
- **Inputs:**
 - Member email or invite code
 - Role change request (member → admin)
 - Removal confirmation
- **Process:** Admin issues invite/add/remove → system updates membership and notifies affected users.
- **Outputs:** Updated member list; notification emails/push.
- **Business rules:** Admin cannot remove last admin without transferring role.
- **Error handling:** Attempting to add existing member yields informative message.

7. Creating Tasks

Purpose: Add tasks to a group (by admin) or personal tasks (by member, if allowed).

- **Actors:** Group Admin (primary), Group Members (if permitted)
- **Preconditions:** User belongs to a group and has create-task permissions.
- **Inputs (task fields):**
 - Task title (required)
 - Task description (required)
 - Assigned medal tier (Gold/Silver/Bronze) → maps to point value
 - Point value (numeric) — either auto-associated with medal or custom
 - Assignee(s) — specific member(s) or "unassigned"
 - Visibility scope (group / subgroup / private)

- Deadline (date/time)
- Recurrence settings (none / daily / weekly / custom)
- Optional note (humorous/instructions)
- Attachments (images, checklists)
- **Process:**
 - User fills task form.
 - System validates fields and business rules.
 - Task saved to group task list; notifications sent to assignee(s)/group if needed.
- **Outputs:** Task record; notification; calendar integration (optional).
- **Postconditions:** Task appears in All Tasks / In Progress / Available depending on status.
- **Business rules:**
 - Only admins can set medals/point weights by default (unless group allows members).
 - Recurring tasks generate future instances per schedule.
- **Error handling:** Missing required fields, invalid date, conflicting assignments.

8. Editing Tasks

Purpose: Update task details or reassign tasks.

- **Actors:** Admin (primary), Original task creator (if permitted)
- **Preconditions:** Task exists and user has permission to edit.
- **Modifiable fields:** All fields except immutable audit fields (creator, created-at).
- **Process:** Save edits → log change in Activity Tab → notify affected users if relevant.
- **Business rules:** Changing assignee should trigger notification; if task was completed, edits should open a new revision record.
- **Error handling:** Conflict resolution if two users edit simultaneously.

9. Claiming Tasks

Purpose: Allow members to claim an unassigned/available task.

- **Actors:** Group Member
- **Preconditions:** Task is in “available” state and not already claimed.
- **Inputs:** Claim action (tap/click)
- **Process:**
 1. User selects claim → system checks availability.
 2. If available, mark task status to *In Progress* and assign user.
 3. Notify group/admin of claim.
- **Outputs:** Task assigned to user; In Progress status; activity log entry.
- **Business rules:** Optionally allow multiple simultaneous claimers (team tasks) if task permits.
- **Error handling:** Inform user if task already claimed, or if they lack permissions.

10. Completing Tasks / Updating Task Status

Purpose: Mark tasks as completed and award points/medals.

- **Actors:** Assigned Member (completer), Admin (for verification)
- **Preconditions:** Task is *In Progress* by the user or assigned to them.
- **Inputs:** Completion action + optional proof (photo, checklist, notes)
- **Process:**
 1. User submits completion.
 2. System records completion and awards configured points/medal.
- **Outputs:** Task status = Completed; points added to user profile; Activity log entry.
- **Business rules:** Some tasks may be auto-approved (low-risk) while others require admin verification.
- **Edge cases:** Reversing a completed task (admin override) should adjust points and leaderboard accordingly.

11. Task Viewing, Filtering & Sorting

Purpose: Provide multiple task views and filters for user convenience.

- **Actors:** All authenticated users

- **Preconditions:** User logged in and belongs to at least one group.
- **Available Views / Filters:**
 - Tabs: All Tasks / In Progress / Completed
 - Date filters: Today / Yesterday / Tomorrow / Custom date range
 - Medal tier / point range
 - Search by keyword
 - Sort by deadline, points, creation date, assignee
- **Outputs:** Filtered & paginated task lists; counts for each category.

12. Leaderboard & Badges

Purpose: Track user standings and milestones to encourage competition and motivation.

- **Actors:** System, Group Members
- **Preconditions:** Points and badge rules configured.
- **Inputs:** Points from task completions, admin-awarded badges, milestone triggers.
- **Process:**
 1. System updates leaderboard in real-time (or at defined intervals).
 2. Badge awarding based on predefined rules (e.g., 10 tasks completed → “Contributor” badge).
- **Outputs:** Leaderboard view; badges on user profile; notifications on earning.
- **Business rules:** Tie-breaking rules for equal points (e.g., earliest to reach points wins).
- **Error handling:** Handle leaderboard caching and eventual consistency without showing stale/incorrect totals.

13. In-App Reward Shop (Admin-Managed)

Purpose: Enable users to redeem accumulated points for rewards defined by the group admin.

- **Actors:** Group Admin (manager), Group Members (buyers)
- **Preconditions:** Shop configured by admin; user has positive points balance.
- **Admin Inputs (shop configuration):**

1. Reward item name
 2. Description
 3. Point cost
 4. Quantity limit (optional)
 5. Type (digital / real / privilege)
 6. Redemption rules (one-time, recurring)
- **User Redemption Inputs:**
 1. Select reward → confirm redemption
 - **Process:**
 1. System checks user's point balance.
 2. Deduct points and create redemption record.
 3. Notify admin to fulfill (if real-world reward) or apply privilege (skip a chore) automatically.
 - **Outputs:** Updated points balance; redemption record; notification to admin and user.
 - **Business rules:** Admin approval required for physical rewards; point refund rules on cancellations.
 - **Error handling:** Insufficient points, item out-of-stock, concurrent redemptions handled atomically.

14. Activity Log / History

Purpose: Provide chronological records for transparency and accountability.

- **Actors:** All users (view limited by role)
- **Recorded events:**
 - Task created/edited/claimed/completed
 - Points awarded/deducted
 - Reward redemptions
 - Member join/leave/promotions
- **Preconditions:** Actions performed are logged.
- **Process:** Append immutable activity entries with timestamp, actor, and metadata.

- **Outputs:** Chronological activity list with filters (by user, task, date range).
- **Business rules:** Retention policy (e.g., keep logs for X years) configurable.
- **Security:** Only authorized users can view full logs; members see group-level logs.

15. AI-Assisted Task Suggestions

Purpose: Use simple AI heuristics to suggest tasks and schedules to users.

- **Actors:** System (AI module), User
- **Preconditions:** Sufficient historical task data; user consents to analytics.
- **Inputs:** User's past tasks, repetition patterns, keywords, time-of-day preferences.
- **Process:**
 1. AI runs offline/batched or real-time analysis.
 2. Generates recommended tasks and suggested schedules.
 3. Presents suggestions in a "Suggested Tasks" list for user/admin to accept or dismiss.
- **Outputs:** Suggested tasks with rationale score and "accept/reject" action.
- **Business rules:** User/in-group opt-in required; suggestions annotated to show they're AI-generated.
- **Error handling / transparency:** Provide explanation (why suggested) and option to turn off AI.

16. Notifications & Reminders

Purpose: Keep users informed of claims, deadlines, approvals, shop changes.

- **Actors:** System, Users
- **Types:**
 - Push notifications
- **Triggers:** New task, task approaching deadline, task claimed, completion verification required, points awarded, shop updates.
- **Inputs:** Notification preferences per user.
- **Business rules:** Time-window for reminders (e.g., no reminders during Do Not Disturb hours unless critical).

- **Error handling:** Queueing & retry for failed push deliveries.

17. Security & Role Management (Functional Aspects)

Purpose: Enforce authentication, authorization, and auditability.

- **Actors:** System Admin, Group Admin
- **Preconditions:** Authentication system in place.
- **Functional elements:**
 - Role-based access control (member, admin, system admin)
 - Session timeout and token revocation
- **Business rules:** Only system admin can perform global user deletion and data export; admins limited to their groups.

18. Integration (Optional functional items)

Purpose: Allow optional integrations that affect functional flows.

- **Possible integrations:**
 - Calendar sync (Google Calendar) — for task deadlines
 - Push notification services (FCM / APNs)
 - OAuth providers (Google, Apple)
 - External reward fulfillment (e-commerce / voucher APIs)
- **Process:** OAuth flows, token storage, periodic sync.
- **Business rules:** User consent required for cross-service data sharing.