RevisionHub Product Requirements Document v8.1

Document Control

Property	Value
Version	8.1
Date	24 January 2026
Status	Active Development
Owner	Product & Engineering

Version History

Version	Date	Summary
v6.0	13 Jan 2026	7-step session model, pre/post confidence, flashcard persistence
v6.1	13 Jan 2026	6-step model (merged reflection+complete), schema corrections
v7.0	15 Jan 2026	Complete consolidation: needs assessment, payload restructure, content schemas, frontend patterns, voice transcription
v8.0	21 Jan 2026	Parent Dashboard v2, Unified Status System, Study Buddy AI, Add Subject with Redistribution, Content Pipeline
v8.1	**24 Jan 2026**	**FEAT-013 Reward System: 7 tables, 17 RPCs, parent config, child catalog, addition requests, Today.tsx refactor**

Document Purpose

PRD v8.1 adds FEAT-013 Reward Configuration & Redemption system, a comprehensive gamification enhancement that connects revision points to real-world rewards. Key additions:

- **Section 16:** Reward Configuration System (FEAT-013)
- **Section 22.3:** New database tables for rewards
- Updated feature completion status

Section 16: Reward Configuration System (FEAT-013)

16.1 Overview

The reward system connects revision points to real-world rewards, creating a collaborative incentive system between parents and children.

Design Principle: "A conversation, not a transaction" - The UI encourages parents to set up rewards together with their child.

16.1.1 Key Features

Feature	Description
Point Weighting	Parents configure how points are calculated (completion, accuracy, focus)
Reward Categories	6 predefined categories with suggested rewards
Custom Rewards	Parents can create custom rewards beyond templates
Redemption Approval	Parent approval flow with auto-approve threshold
Addition Requests	Children can request rewards from the catalog
Reward Limits	Per-day, per-week, per-month limits on rewards

16.2 Database Schema

16.2.1 reward_categories

Reference table for reward category types.

```sql
CREATE TABLE reward_categories (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  code TEXT UNIQUE NOT NULL,
  name TEXT NOT NULL,
  icon TEXT NOT NULL,
  display_order INTEGER DEFAULT 0,
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMPTZ DEFAULT now()
);
```

Seed Data:

code	name	icon
screen_time	Screen Time	📱
treats	Treats	🍦
activities	Activities	🎯
pocket_money	Pocket Money	💰
privileges	Privileges	⭐
custom	Custom	🎁

16.2.2 reward_templates

Suggested rewards per category that parents can enable.

```sql
CREATE TABLE reward_templates (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  category_id UUID REFERENCES reward_categories(id),
  name TEXT NOT NULL,
  suggested_points INTEGER NOT NULL,
  display_order INTEGER DEFAULT 0,
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMPTZ DEFAULT now()
);
```

Example Seed Data:

category	name	suggested_points
screen_time	15 minutes extra gaming	100
screen_time	30 minutes extra gaming	200
screen_time	1 hour extra screen time	350
treats	Small treat	150
treats	Medium treat	300
activities	Cinema trip	800
pocket_money	£5	500
privileges	Stay up 30 mins late	200

16.2.3 child_rewards

Parent-enabled rewards for a specific child.

```sql
CREATE TABLE child_rewards (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  child_id UUID NOT NULL REFERENCES children(id) ON DELETE CASCADE,
  category_id UUID NOT NULL REFERENCES reward_categories(id),
  template_id UUID REFERENCES reward_templates(id),
  name TEXT NOT NULL,
  emoji TEXT DEFAULT '🎁',
  points_cost INTEGER NOT NULL,
  limit_type TEXT CHECK (limit_type IN ('per_day', 'per_week', 'per_month', 'unlimited')),
  limit_count INTEGER,
  is_active BOOLEAN DEFAULT true,
  created_at TIMESTAMPTZ DEFAULT now(),
  updated_at TIMESTAMPTZ DEFAULT now()
);
```

16.2.4 child_point_config

Point weighting configuration per child.

```sql
```

```sql
CREATE TABLE child_point_config (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  child_id UUID UNIQUE NOT NULL REFERENCES children(id) ON DELETE CASCADE,
  mode TEXT DEFAULT 'auto' CHECK (mode IN ('auto', 'manual')),
  completion_weight INTEGER DEFAULT 40,
  accuracy_weight INTEGER DEFAULT 35,
  focus_weight INTEGER DEFAULT 25,
  auto_approve_threshold INTEGER DEFAULT 0,
  created_at TIMESTAMPTZ DEFAULT now(),
  updated_at TIMESTAMPTZ DEFAULT now(),

  CONSTRAINT weights_sum_100 CHECK (
    mode = 'auto' OR (completion_weight + accuracy_weight + focus_weight = 100)
  ),
  CONSTRAINT weights_min_10 CHECK (
    mode = 'auto' OR (
      completion_weight >= 10 AND accuracy_weight >= 10 AND focus_weight >= 10
    )
  )
);
```

16.2.5 reward_redemptions

Child requests to redeem rewards.

```sql
CREATE TABLE reward_redemptions (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  reward_id UUID NOT NULL REFERENCES child_rewards(id) ON DELETE CASCADE,
  status TEXT DEFAULT 'pending' CHECK (
    status IN ('pending', 'approved', 'declined', 'cancelled', 'expired')
  ),
  points_spent INTEGER NOT NULL,
  requested_at TIMESTAMPTZ DEFAULT now(),
  resolved_at TIMESTAMPTZ,
  resolved_by UUID REFERENCES auth.users(id),
  decline_reason TEXT,
  expires_at TIMESTAMPTZ DEFAULT (now() + INTERVAL '7 days')
);
```

16.2.6 reward_addition_requests

Child requests for parents to add new rewards from the catalog.

```sql
CREATE TABLE reward_addition_requests (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  child_id UUID NOT NULL REFERENCES children(id) ON DELETE CASCADE,
  template_id UUID NOT NULL REFERENCES reward_templates(id) ON DELETE CASCADE,
  status TEXT NOT NULL DEFAULT 'pending' CHECK (
    status IN ('pending', 'approved', 'declined')
  ),
  requested_at TIMESTAMPTZ NOT NULL DEFAULT now(),
  resolved_at TIMESTAMPTZ,
  resolved_by UUID REFERENCES auth.users(id),
  parent_note TEXT,
  created_at TIMESTAMPTZ NOT NULL DEFAULT now()
);

-- Prevent duplicate pending requests
CREATE UNIQUE INDEX idx_unique_pending_addition
  ON reward_addition_requests(child_id, template_id)
  WHERE status = 'pending';
```

16.3 RPC Functions

16.3.1 Configuration RPCs

rpc_get_child_reward_config

Returns complete reward configuration for a child.

```sql
```

```
rpc_get_child_reward_config(p_child_id UUID)
RETURNS JSON {
  point_config: {
    mode: 'auto' | 'manual',
    completion_weight: INTEGER,
    accuracy_weight: INTEGER,
    focus_weight: INTEGER,
    auto_approve_threshold: INTEGER
  },
  categories: [{
    id: UUID,
    code: TEXT,
    name: TEXT,
    icon: TEXT
  }],
  rewards: [{
    id: UUID,
    category_code: TEXT,
    name: TEXT,
    emoji: TEXT,
    points_cost: INTEGER,
    limit_type: TEXT,
    limit_count: INTEGER,
    is_active: BOOLEAN
  }],
  points_balance: INTEGER,
  pending_count: INTEGER
}
```

rpc_save_point_config

Saves point weighting configuration.

```sql
rpc_save_point_config(
  p_child_id UUID,
  p_mode TEXT,
  p_completion_weight INTEGER,
  p_accuracy_weight INTEGER,
  p_focus_weight INTEGER,
  p_auto_approve INTEGER
)
RETURNS JSON { success: BOOLEAN, error?: TEXT }
```

rpc_upsert_child_reward

Creates or updates a reward for a child.

```sql
rpc_upsert_child_reward(
  p_child_id UUID,
  p_reward_id UUID,          -- NULL for new
  p_category_id UUID,
  p_name TEXT,
  p_points_cost INTEGER,
  p_emoji TEXT,
  p_limit_type TEXT,
  p_limit_count INTEGER
)
RETURNS JSON { success: BOOLEAN, reward_id: UUID }
```

rpc_remove_child_reward

Deletes a reward (soft delete via is_active = false).

```sql
rpc_remove_child_reward(p_reward_id UUID)
RETURNS JSON { success: BOOLEAN }
```

rpc_enable_template_rewards

Bulk-enables rewards from templates.

```sql
rpc_enable_template_rewards(
  p_child_id UUID,
  p_template_ids UUID[]
)
RETURNS JSON { success: BOOLEAN, created_count: INTEGER }
```

16.3.2 Redemption RPCs

rpc_request_reward_redemption

Child requests to redeem a reward.

```sql
```

```
rpc_request_reward_redemption(
  p_child_id UUID,
  p_reward_id UUID
)
RETURNS JSON {
  success: BOOLEAN,
  redemption_id?: UUID,
  auto_approved?: BOOLEAN,
  error?: TEXT  -- 'insufficient_points' | 'limit_reached' | 'reward_inactive'
}
```

Logic:

1. Check points balance
2. Check reward limits for period
3. If auto_approve_threshold met → approve immediately
4. Else → create pending request

rpc_resolve_redemption

Parent approves or declines a redemption request.

```sql
rpc_resolve_redemption(
  p_redemption_id UUID,
  p_action TEXT,             -- 'approve' | 'decline'
  p_reason TEXT              -- optional decline reason
)
RETURNS JSON { success: BOOLEAN }
```

rpc_get_pending_redemptions

Returns pending redemptions for parent approval.

```sql
rpc_get_pending_redemptions(p_parent_id UUID)
RETURNS JSON [{
  id: UUID,
  child_id: UUID,
  child_name: TEXT,
  reward_name: TEXT,
  emoji: TEXT,
  points_cost: INTEGER,
  requested_at: TIMESTAMPTZ,
  expires_at: TIMESTAMPTZ,
  child_current_balance: INTEGER
}]
```

rpc_cancel_redemption_request

Child cancels their own pending request.

```sql
rpc_cancel_redemption_request(p_redemption_id UUID)
RETURNS JSON { success: BOOLEAN }
```

16.3.3 Child Catalog RPCs

rpc_get_child_rewards_catalog

Returns rewards with availability status.

```sql
rpc_get_child_rewards_catalog(p_child_id UUID)
RETURNS JSON {
  points_balance: INTEGER,
  rewards: [{
    id: UUID,
    name: TEXT,
    emoji: TEXT,
    category_code: TEXT,
    category_name: TEXT,
    points_cost: INTEGER,
    can_afford: BOOLEAN,
    is_available: BOOLEAN,
    limit_type: TEXT,
    times_used_in_period: INTEGER
  }]
}
```

rpc_get_child_rewards_summary

Returns summary for mini dashboard card.

```sql
```

```
rpc_get_child_rewards_summary(p_child_id UUID)
RETURNS JSON {
  points_balance: INTEGER,
  unlocked_count: INTEGER,
  next_reward_name: TEXT,
  next_reward_points: INTEGER,
  points_needed: INTEGER
}
```

rpc_get_child_rewards_dashboard

Returns full dashboard stats for hero card.

```sql
rpc_get_child_rewards_dashboard(p_child_id UUID)
RETURNS JSON {
  points_balance: INTEGER,
  total_earned: INTEGER,
  total_spent: INTEGER,
  available_rewards: INTEGER,
  unlocked_count: INTEGER,
  pending_redemptions: INTEGER,
  pending_additions: INTEGER,
  total_redeemed: INTEGER
}
```

16.3.4 Addition Request RPCs

rpc_request_reward_addition

Child requests a reward be added from the catalog.

```sql
rpc_request_reward_addition(
  p_child_id UUID,
  p_template_id UUID
)
RETURNS JSON {
  success: BOOLEAN,
  request_id?: UUID,
  error?: TEXT  -- 'already_have' | 'already_requested'
}
```

rpc_resolve_addition_request

Parent approves or declines an addition request.

```sql
rpc_resolve_addition_request(
  p_request_id UUID,
  p_action TEXT,              -- 'approve' | 'decline'
  p_parent_note TEXT,
  p_points_cost INTEGER       -- override suggested points
)
RETURNS JSON { success: BOOLEAN, reward_id?: UUID }
```

On Approve:

1. Creates child_reward from template
2. Uses p_points_cost or template.suggested_points
3. Marks request as approved

rpc_get_pending_addition_requests

Returns pending addition requests for parent.

```sql
rpc_get_pending_addition_requests(p_parent_id UUID)
RETURNS JSON [{
  id: UUID,
  child_id: UUID,
  child_name: TEXT,
  template_id: UUID,
  template_name: TEXT,
  category_name: TEXT,
  suggested_points: INTEGER,
  requested_at: TIMESTAMPTZ
}]
```

rpc_get_reward_catalog_for_child

Returns full template catalog with status.

```sql
```

```
rpc_get_reward_catalog_for_child(p_child_id UUID)
RETURNS JSON [{
  id: UUID,
  name: TEXT,
  suggested_points: INTEGER,
  category_code: TEXT,
  category_name: TEXT,
  category_icon: TEXT,
  is_added: BOOLEAN,
  child_reward_id: UUID,
  request_pending: BOOLEAN,
  pending_request_id: UUID
}]
```

16.4 User Interface

16.4.1 Parent: RewardManagement Page

Route: `/parent/rewards`

Layout:

```
┌─────────────────────────────────────────────────────┐
│ [Child Tabs: Hannah | Toby]                         │
├─────────────────────────────────────────────────────┤
│ Hero Card                                           │
│ Points: 245 │ Rewards: 8 │ Pending: 2 │ Redeemed: 5 │
├─────────────────────────────────────────────────────┤
│ 🔔 Pending Redemptions (2)         [Approve/Decline]│
├─────────────────────────────────────────────────────┤
│ 📩 Reward Requests (1)             [Add/Decline]    │
├─────────────────────────────────────────────────────┤
│ Point Weighting                    [Edit]           │
│ Mode: Manual │ Completion: 50% │ Accuracy: 30% │ Focus: 20% │
├─────────────────────────────────────────────────────┤
│ Active Rewards                                      │
│ 📱 Screen Time (3)                 [Manage]         │
│ 🍦 Treats (2)                      [Manage]         │
│ 🎯 Activities (4)                  [Manage]         │
│                                                     │
│ [+ Add Custom Reward]                               │
└─────────────────────────────────────────────────────┘
```

16.4.2 Child: ChildRewardsCatalog Page

Route: `/child/rewards`

Layout:

```
┌─────────────────────────────────────────────────────┐
│ My Rewards 🎁                                       │
├─────────────────────────────────────────────────────┤
│ Hero Card (gradient)                                │
│ ⭐ 245 Points │ Earned: 1,240 │ Unlocked: 3 │ Redeemed: 5 │
├─────────────────────────────────────────────────────┤
│ ┌─ Pending Redemptions ─┐ ┌─ Requests Sent ──────┐  │
│ │ 🍦 Ice cream - 50pts  │ │ 📺 Extra TV time     │  │
│ │ [Cancel]              │ │ Waiting for parent   │  │
│ └───────────────────────┘ └──────────────────────┘  │
├─────────────────────────────────────────────────────┤
│ [ My Rewards (6) ] [ Browse Catalog (24) ] [ History (5) ] │
├─────────────────────────────────────────────────────┤
│ Ready to Claim! ✓                                   │
│ ┌────────┐ ┌────────┐ ┌────────┐                    │
│ │ 🎮     │ │ 🍿     │ │ 💰     │                    │
│ │ Gaming │ │ Movie  │ │ £5     │                    │
│ │ 100pts │ │ 75pts  │ │ 200pts │                    │
│ └────────┘ └────────┘ └────────┘                    │
├─────────────────────────────────────────────────────┤
│ Keep Working Towards                                │
│ ┌────────┐ ┌────────┐                               │
│ │ 🎬     │ │ 🎉     │ Progress bars showing         │
│ │ Cinema │ │ Party  │ points needed                 │
│ │ 800pts │ │ 1500pt │                               │
│ └────────┘ └────────┘                               │
└─────────────────────────────────────────────────────┘
```

16.4.3 Today.tsx Component Architecture

The Today page was refactored to extract components:

```
Today.tsx (orchestrator)
├── TodayHeader.tsx       - Greeting + streak badge
├── SessionList.tsx       - Today's sessions card
├── TodayProgressCard.tsx - Week progress grid
├── UpcomingSection.tsx   - Coming up timeline
├── StreakMomentumCard.tsx - Streak card (half-width)
├── RewardsMiniCard.tsx   - Rewards summary (half-width)
├── TodayTipCard.tsx      - Daily tip rotation
└── EmptyState.tsx        - Loading/Error/Empty states
```

16.5 Business Rules

Rule	Implementation
Points held during pending redemption	Not yet implemented
7-day expiry on pending redemptions	Requires cron job
Auto-approve threshold	If points_cost ≤ threshold, approve immediately
Reward limits	Checked before allowing redemption
Duplicate addition requests	Prevented by unique index
Deleted reward with pending redemption	Auto-decline with system message

---

16.6 UI Components

16.6.1 Parent Components

Component	Location	Purpose
`RewardManagement`	`pages/parent/`	Main management page
`PointWeightingConfig`	`components/parent/rewards/`	Slider configuration
`RewardEditor`	`components/parent/rewards/`	Add/edit modal
`PendingRedemptions`	`components/parent/rewards/`	Redemption approval queue
`PendingAdditionRequests`	`components/parent/rewards/`	Addition request queue

16.6.2 Child Components

Component	Location	Purpose
`ChildRewardsCatalog`	`pages/child/`	Full rewards page with tabs
`RewardsMiniCard`	`components/child/today/`	Dashboard mini card
`RedemptionModal`	`components/child/rewards/`	Request confirmation
`RewardToast`	`components/child/rewards/`	Success notification

16.6.3 Navigation Updates

Location	Change
`ChildNav.tsx`	Added "Rewards" link
`AppHeader.tsx`	Added points badge for child users
`App.tsx`	Added `/child/rewards` route

Updated Section 18: Feature Completion Status

18.1 Completed Features

Feature	ID	Date	Notes
6-Step Session Model	v6.1	13 Jan 2026	Merged reflection into complete
Session Reflections	FEAT-001	14 Jan 2026	Voice notes + transcription
Atomic Session Start	FEAT-002	14 Jan 2026	Race condition fix
Pilot Seed Content	FEAT-003	14 Jan 2026	68 content units, 4 topics
Language Level Extraction	FEAT-004	14 Jan 2026	Policy → content level
Payload Restructure	FEAT-005	15 Jan 2026	Flashcards → Recall, arrays
RecallStep Refactor	FEAT-007	15 Jan 2026	Component extraction pattern
Parent Insights Dashboard	FEAT-008	15 Jan 2026	7 RPCs, 10 widgets, AI advice
Parent Dashboard Redesign	FEAT-009	15 Jan 2026	Multi-child support, HeroStatusBanner

Feature	ID	Date	Notes
Unified Status System	FEAT-010	16 Jan 2026	4 status levels, helper function
Study Buddy AI	FEAT-011	16 Jan 2026	Text + voice, 6 phases complete
Add Subject Redistribution	FEAT-012	16 Jan 2026	Priority reordering, impact assessment
Reward System Phase 1-3	**FEAT-013**	**24 Jan 2026**	**7 tables, 17 RPCs, parent+child UI, addition requests**

Updated Section 22: Appendix Database Schema Summary

22.3 Reward Tables (App Database)

Table	Purpose
`reward_categories`	6 predefined reward category types
`reward_templates`	Suggested rewards per category
`child_rewards`	Parent-enabled rewards per child
`child_point_config`	Point weighting configuration
`reward_redemptions`	Child redemption requests
`reward_addition_requests`	Child requests for new rewards

Updated Section 21: Related Documents

Document	Purpose
BOARD_DOCUMENT_CAPTURE_STRATEGY_v2_0.md	Content ingestion pipeline (separate)
FEAT-010_Unified_Status_System_v2.0.md	Status system specification
FEAT-011_Study_Buddy_AI_Chatbot.md	Study Buddy full specification
FEAT-012_Add_Subject_With_Redistribution.md	Subject management specification
FEAT-013_Reward_Configuration_v2.md	**Reward system specification**
RevisionHub_Production_Backlog.md	Technical debt tracking
RevisionHub_Pilot_Seed_Content_Blueprint.md	Content scaling blueprint

---

Document End

PRD v8.1 - 24 January 2026 Additions: FEAT-013 Reward System (Section 16), database schema updates, feature completion