**Appendix A: Text-Based Wireframes**

**Note:** Each block below represents a full screen. Elements in square brackets (e.g., [Logo]) indicate UI components. Use these as the basis for visual mockups.

**A.1 Sign-Up & Privacy Flow**

**A.1.1 Sign-Up Landing Screen**

css

CopyEdit

┌────────────────────────────────────────────────────┐

│ [App Logo]

│ │

│ Welcome to AppForge.AI!

│ Discover micro-games shaping tomorrow.

│ │

│ [ Sign Up with Email ]

│ │

│ Already have an account? [ Log In ]

│ │

└────────────────────────────────────────────────────┘

* **[App Logo]**: Centered at top.
* **Tagline**: Two lines, centered.
* **“Sign Up with Email”**: Primary button (centered).
* **“Log In”**: Secondary link below.

**A.1.2 Email & Password Input**

markdown

CopyEdit

┌────────────────────────────────────────────────────┐

│ [←] Create Your Account

│ │

│ Email Address:

│ [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_]

│ │

│ Create Password (min 8 chars):

│ [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_]

│ │

│ [ Continue ] (disabled until valid inputs)

│ │

│ Already have an account? [ Log In ]

│ │

└────────────────────────────────────────────────────┘

* **Back Arrow [←]**: Top-left to return.
* **Email & Password Fields**: Centered.
* **Continue**: Disabled until valid.
* **Log In**: Bottom link.

**A.1.3 Privacy & Consent**

css

CopyEdit

┌────────────────────────────────────────────────────┐

│ [←] Your Privacy Matters │

│ │

│ We collect anonymized data to improve │

│ your experience. You can control what we collect │

│ below. │

│ │

│ [✔] Analytics & Performance │

│ “Collect gameplay data for improvements.” │

│ │

│ [ ] Personalization │

│ “Provide personalized suggestions.” │

│ │

│ [ ] Marketing Communications │

│ “Send occasional email updates/offers.” │

│ │

│ [ Learn More ] (opens full policy modal) │

│ │

│ [ Accept & Continue ] │

│ (enabled only if Analytics is checked) │

│ │

└────────────────────────────────────────────────────┘

* **Title**: “Your Privacy Matters.”
* **Description**: Short explanation.
* **Checkboxes/Toggles**:
  + Analytics & Performance (default ✓).
  + Personalization (empty).
  + Marketing Communications (empty).
* **“Learn More”**: Opens full policy in modal.
* **“Accept & Continue”**: Enabled only when Analytics is checked.

**A.1.4 Privacy Policy Modal**

pgsql

CopyEdit

┌────────────────────────────────────────────────────┐

│ [X] Privacy Policy │

│ │

│ We collect the following: │

│ - Gameplay events (taps, scores, shares) │

│ - Device & performance metrics │

│ - Feedback you submit (anonymized) │

│ │

│ Why we collect: │

│ - Improve game balance & suggestions │

│ - Detect abusive content │

│ - Offer personalized features │

│ │

│ You can: │

│ - Opt out anytime in Settings → Privacy │

│ - Request data export or deletion via our │

│ “Right to Know/Erase” page │

│ │

│ [ Scroll through full policy text… ] │

│ │

│ [ I Understand, Close ] │

│ │

└────────────────────────────────────────────────────┘

* **[X]**: Closes modal.
* **Scrollable Area**: Full policy text.
* **“I Understand, Close”**: Closes and returns, preserving toggles.

**A.1.5 Confirmation Screen**

sql

CopyEdit

┌────────────────────────────────────────────────────┐

│ │

│ ✅ You’re all set! │

│ │

│ Thank you for joining AppForge.AI. │

│ Your preferences have been saved. │

│ │

│ [ Get Started ] │

│ │

└────────────────────────────────────────────────────┘

* **Green Check Icon**: Centered.
* **Confirmation Text**: Two lines.
* **“Get Started”**: Button to proceed.

**A.1.6 Onboarding Overlay (After Consent)**

vbnet

CopyEdit

┌────────────────────────────────────────────────────┐

│ [Skip Tutorial] │

│ │

│ Welcome to Micro-Games! │

│ │

│ • Tap to play. │

│ • Watch short ads to earn bonus points. │

│ • Submit feedback to shape our next apps. │

│ │

│ [ Got It, Let’s Play ] │

│ │

└────────────────────────────────────────────────────┘

* **“Skip Tutorial”**: Top-right.
* **Bullets**: Basic instructions.
* **“Got It, Let’s Play”**: Button at bottom.

**A.2 Micro-Game Core Screens**

**A.2.1 Main Gameplay Screen (Tapper Example)**

less

CopyEdit

┌────────────────────────────────────────────────────┐

│ [≡ Menu] Ghost Threads Challenge [Share ↗] │

│ │

│ [ Background Theme Art (semi-transparent) ] │

│ │

│ [ Large Circular Tap Button ] │

│ │

│ Timer: 00:20 Score: 850 [XP Bar] │

│ │

│ [ 🎥 Watch Ad ] [ ✍ Feedback ] │

│ │

│ [Home] [Feed] [Profile] [Settings] │

└────────────────────────────────────────────────────┘

* **Header**:
  + Left: Menu icon opens side drawer.
  + Center: Title “Ghost Threads Challenge.”
  + Right: Share icon.
* **Tap Button**: Centered.
* **Footer Row**: Timer, Score, XP bar.
* **Side Controls**: Watch Ad (bottom right) and Feedback (bottom left).
* **Bottom Nav**: Four icons.

**A.2.2 Post-Game Summary Modal**

mathematica

CopyEdit

┌────────────────────────────────────────────────────┐

│ [Overlay: semi-transparent dark] │

│ ┌──────────────────────────────────────────────┐ │

│ │ Game Over! │ │

│ │ │ │

│ │ Final Score: 1,250 │ │

│ │ Best Combo: 20 taps in 3 seconds │ │

│ │ Rank: Top 15% today │ │

│ │ │ │

│ │ [ Share ] [ Play Again ] │ │

│ │ │ │

│ │ [ Exit ] (link style) │ │

│ └──────────────────────────────────────────────┘ │

│ │

└────────────────────────────────────────────────────┘

* **Overlay**: Blocks background.
* **Stats Card**: White, rounded.
* **Buttons**: “Share,” “Play Again,” “Exit.”

**A.2.3 Feedback Modal**

markdown

CopyEdit

┌────────────────────────────────────────────────────┐

│ [Overlay: semi-transparent dark] │

│ ┌──────────────────────────────────────────────┐ │

│ │ Help Us Improve │ │

│ │ │ │

│ │ [ 👍 ] [ 👎 ] │ │

│ │ “Enjoyed?” “Needs Work?” │ │

│ │ │ │

│ │ Comments (optional): │ │

│ │ [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_] │ │

│ │ │ │

│ │ [ Submit ] [ Skip ] │ │

│ │ │ │

│ └──────────────────────────────────────────────┘ │

│ │

└────────────────────────────────────────────────────┘

* **Header**: “Help Us Improve.”
* **Icons**: Thumbs up/down.
* **Comments Field**: Single- or multi-line.
* **Buttons**: “Submit” (enabled after thumbs tap), “Skip.”

**A.2.4 Side Drawer Menu**

pgsql

CopyEdit

┌────────────────────────────────────────────────────┐

│ [User Avatar] Jasmine Douglas │

│ ┌────────────────────────────────────────────┐ │

│ │ • My Profile │ │

│ │ • Settings │ │

│ │ • Support │ │

│ │ • Log Out │ │

│ └────────────────────────────────────────────┘ │

│ │

└────────────────────────────────────────────────────┘

* **Avatar & Name**: Top.
* **Menu Items**: Four tappable rows.

**A.2.5 Settings Page**

less

CopyEdit

┌────────────────────────────────────────────────────┐

│ [←] Settings │

│ │

│ Account │

│ • Email: jasmine@example.com │

│ • Change Password │

│ │

│ Notifications │

│ • [✔] Push Notifications (toggle) │

│ │

│ Privacy │

│ • [✔] Analytics & Performance (toggle) │

│ • [ ] Personalization (toggle) │

│ • [ ] Marketing Communications (toggle) │

│ │

│ About │

│ • Version 1.0.3 │

│ • Terms of Service │

│ • Privacy Policy │

│ │

│ [ Delete My Account ] (red text) │

│ │

└────────────────────────────────────────────────────┘

* **Back Arrow**: Top-left.
* **Sections**: Account, Notifications, Privacy, About.
* **Delete My Account**: Red text at bottom.

**A.3 Layer 2: Curator Dashboard (Internal Users)**

**A.3.1 Curator Login**

markdown

CopyEdit

┌────────────────────────────────────────────────────┐

│ [App Logo] │

│ │

│ Curator Portal │

│ │

│ Email Address: │

│ [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_] │

│ │

│ Password: │

│ [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_] │

│ │

│ [ Log In ] │

│ │

│ Forgot Password? │

│ │

└────────────────────────────────────────────────────┘

* Simple login form with email/password fields and “Log In” button.
* “Forgot Password?” link below.

**A.3.2 Dashboard Home (Channel Overview)**

less

CopyEdit

┌────────────────────────────────────────────────────┐

│ [Logo] Channel Health Dashboard [📅] │

│ │

│ Filter: [ Last 7 Days ▼ ] [ New Briefs: 5 ] │

│ │

│ ┌──────────────────────────────────────────────┐ │

│ │ # | Channel Name | Health | Growth% │ │

│ │ 1 | Ghost Drama | 87 | +12% │ │

│ │ 2 | Neon Nostalgia | 82 | +8% │ │

│ │ 3 | Midnight Overthink | 76 | +5% │ │

│ │ … | … | … | … │ │

│ └──────────────────────────────────────────────┘ │

│ │

│ [ Search Channel… ] [ Approve Selected ] │

│ │

│ [ Prev ] [ 1 ] [ 2 ] [ 3 ] [ Next ] │

│ │

└────────────────────────────────────────────────────┘

* **Header**: Logo left; date filter icon right.
* **Filter Bar**: Dropdown for date, “New Briefs” count.
* **Table**: Paginated list (20 rows), columns: #, Channel Name, Health Score, Growth%.
* **Actions**: “Search Channel…”, “Approve Selected.”
* **Pagination**: Prev/Next.

**A.3.3 Channel Detail Page**

yaml

CopyEdit

┌────────────────────────────────────────────────────┐

│ [←] Dashboard > Channel Detail > Ghost Drama │

│ │

│ Title: Ghost Drama │

│ Health Score: [ 87 / 100 (circular gauge) ] │

│ Last Updated: May 30, 2025 │

│ [ Approve ] [ Reject ] │

│ │

│ ┌──────────────────────────────────────────────┐ │

│ │ Metric │ Value │ Trend │ Change │ │

│ │ Plays │ 12,345 │ ↑ +8% │ │ │

│ │ Shares │ 4,567 │ ↓ −2% │ │ │

│ │ Sentiment│ 92% Pos │ → ±0% │ │ │

│ └──────────────────────────────────────────────┘ │

│ │

│ Top Micro-Games: │

│ • Ghost Tapper – Plays: 8,901 | Shares: 1,234 │

│ • Brief Drama – Plays: 3,444 | Shares: 456 │

│ │

│ User Feedback Word Cloud: │

│ [ word cloud graphic placeholder ] │

│ │

│ Asset Suggestions: │

│ • Art Style: Neon Retro [thumbnail] │

│ • Audio Mood: “70s Synthwave” [▶ play clip] │

│ │

│ ┌──────────────────────────────────────────────┐ │

│ │ Channel Brief JSON (editable) │ │

│ │ { │ │

│ │ "channel\_id": "ghost\_drama", │ │

│ │ "title": "Ghost Drama", │ │

│ │ "description": "Tap to uncover hidden souls", ││

│ │ "art\_style\_tags": ["neon", "retro"], │ │

│ │ … │ │

│ │ } │ │

│ └──────────────────────────────────────────────┘ │

│ │

│ [ Save Changes ] [ Publish & Build Prototype ] │

│ │

└────────────────────────────────────────────────────┘

* **Breadcrumbs**: Top.
* **Header**: Title, Health Score gauge, last update date, “Approve”/“Reject.”
* **Metrics Table**: Plays, Shares, Sentiment with trend arrows.
* **Top Micro-Games**: List of two with stats.
* **Word Cloud**: Placeholder.
* **Asset Suggestions**: Thumbnail + play preview.
* **Channel Brief JSON**: Editable code block.
* **Actions**: “Save Changes” and “Publish & Build Prototype.”

**A.4 Layer 3: Complex App User Flows**

**A.4.1 Complex App Selection Screen**

css

CopyEdit

┌────────────────────────────────────────────────────┐

│ [Logo] Select Your App │

│ │

│ ┌───────────────────────┐ ┌─────────────────────┐ │

│ │ Breakup Bootcamp │ │ Loopr Pro │ │

│ │ (icon + tagline) │ │ (icon + tagline) │ │

│ └───────────────────────┘ └─────────────────────┘ │

│ │

│ ┌───────────────────────┐ ┌─────────────────────┐ │

│ │ Wellness Sphere │ │ Future Fiesta │ │

│ │ (icon + tagline) │ │ (icon + tagline) │ │

│ └───────────────────────┘ └─────────────────────┘ │

│ │

│ [ Refresh Apps ] │

│ │

└────────────────────────────────────────────────────┘

* Four app tiles (two per row), each with icon and tagline.
* **“Refresh Apps”**: Button to update list.

**A.4.2 Complex App Welcome & Onboarding**

pgsql

CopyEdit

┌────────────────────────────────────────────────────┐

│ [←] Welcome to Breakup Bootcamp │

│ │

│ “Hey Jasmine, ready to transform your loops?” │

│ │

│ [10-sec looping background video or animation] │

│ │

│ 1. Personal Info: │

│ • Age Range: [ 18-24 ▼ ] │

│ • Goals (select all that apply): │

│ [✔] Move On [ ] Find Closure │

│ [ ] Build Confidence │

│ │

│ [ Next ] │

│ │

└────────────────────────────────────────────────────┘

* **Back Arrow**: Top-left.
* **Personal Info Fields**: Age range dropdown, goal checkboxes.
* **“Next”**: Button at bottom.

**A.4.3 Mood Baseline (Onboarding)**

less

CopyEdit

┌────────────────────────────────────────────────────┐

│ [←] Mood Baseline │

│ │

│ “How are you feeling today?” │

│ │

│ Mood Slider (0–10): │

│ [ 😭 ] — [ 😐 ] — [ 😊 ] │

│ 0 5 10 │

│ │

│ [ Next ] │

│ │

└────────────────────────────────────────────────────┘

* **Mood Slider**: With emoji anchors at 0, 5, 10.
* **“Next”**: Proceeds to interests.

**A.4.4 Interests Selection (Onboarding)**

css

CopyEdit

┌────────────────────────────────────────────────────┐

│ [←] Select Your Interests │

│ │

│ “What features interest you most?” │

│ │

│ [✔] Daily Affirmations [ ] Community Support│

│ [ ] Challenges [ ] AI Chat Coach │

│ [ ] Habit Tracker │

│ │

│ [ Finish & Enter App ] │

│ │

└────────────────────────────────────────────────────┘

* **Checkbox Grid**: Four interest options.
* **“Finish & Enter App”**: Completes onboarding.

**A.4.5 Complex App Main Dashboard (Breakup Bootcamp Example)**

less

CopyEdit

┌────────────────────────────────────────────────────┐

│ Breakup Bootcamp [⚙ Settings] [🔔 Alerts] │

│ │

│ “Good morning, Jasmine. Ready for today’s │

│ Breakthrough Challenge?” │

│ │

│ ┌──────────────────────────────────────────────┐ │

│ │ Today’s Challenge │ │

│ │ Title: Find Closure │ │

│ │ Progress: [●●●○○] (3/5 steps complete) │ │

│ │ [ Start Now ] │ │

│ └──────────────────────────────────────────────┘ │

│ │

│ ┌──────────────────────────────────────────────┐ │

│ │ Recent Progress │ │

│ │ • Completed “Reflect on the Past” (May 28) │ │

│ │ • Joined “Healing Circle” (May 27) │ │

│ │ • Mood Baseline: 7 (May 28) │ │

│ └──────────────────────────────────────────────┘ │

│ │

│ ┌──────────────────────────────────────────────┐ │

│ │ Community Highlights │ │

│ │ “I finished Day 1 and feel more confident.” │ │

│ │ “This group helped me see things differently.”│ │

│ │ [ See More ] │ │

│ └──────────────────────────────────────────────┘ │

│ │

│ [ Home ] [ Explore ] [ Chat ] [ Profile ] │

│ │

└────────────────────────────────────────────────────┘

* **Header**: App name, Settings, Notifications icons.
* **Daily Challenge Card**: Title, progress bar, “Start Now.”
* **Recent Progress**: List of last actions.
* **Community Highlights**: Snippet of community posts with “See More” link.
* **Bottom Nav**: Four icons.

**A.4.6 Challenge Execution Screen**

vbnet

CopyEdit

┌────────────────────────────────────────────────────┐

│ [←] Find Closure │

│ │

│ Step 2 of 5: Choose a coping strategy │

│ │

│ ┌──────────────────────────────────────────────┐ │

│ │ “When you feel overwhelmed, take a deep │ │

│ │ breath and write down three things you are │ │

│ │ grateful for.” │ │

│ └──────────────────────────────────────────────┘ │

│ │

│ Choices: │

│ ( ) Guided deep-breathing exercise │

│ ( ) Write gratitude list │

│ ( ) Listen to a short ambient track │

│ │

│ [ Submit ] │

│ │

└────────────────────────────────────────────────────┘

* **Breadcrumb**: Back arrow + step indicator.
* **Instruction Box**: Text from GPT.
* **Choices**: Radio options.
* **Submit**: Button to advance.

**A.4.7 AI Chat Coach Screen**

less

CopyEdit

┌────────────────────────────────────────────────────┐

│ Breakup Bootcamp [←] AI Coach │

│ │

│ ┌──────────────────────────────────────────────┐ │

│ │ [👤 User Avatar] Hi, I’m here for you. │ │

│ └──────────────────────────────────────────────┘ │

│ │

│ ┌──────────────────────────────────────────────┐ │

│ │ User: “I’m feeling lost and anxious.” │ │

│ │ ─────────── │ │

│ │ Coach: “I understand. Let’s try a grounding │ │

│ │ exercise together. Can you name 3 things │ │

│ │ you can see right now?” │ │

│ └──────────────────────────────────────────────┘ │

│ │

│ [ Type your message… \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ] [Send] │

│ │

└────────────────────────────────────────────────────┘

* **Header**: App name, Back arrow, “AI Coach” label.
* **Chat Bubbles**: Alternating user/coach text.
* **Input Field**: Text area + “Send” button at bottom.

**A.5 Error & Empty States**

**A.5.1 Network Error Toast**

pgsql

CopyEdit

┌────────────────────────────────────────────────────┐

│ ⚠ Unable to connect. Please check your internet │

│ connection and try again. [ Retry ] │

└────────────────────────────────────────────────────┘

* Transient toast at bottom or top.

**A.5.2 Empty Channel List (Curator)**

pgsql

CopyEdit

┌────────────────────────────────────────────────────┐

│ No channels found for this period. │

│ Try adjusting your filter or wait for new data. │

│ │

│ [ Refresh ] │

└────────────────────────────────────────────────────┘

**A.5.3 No Challenges Available (Complex App)**

pgsql

CopyEdit

┌────────────────────────────────────────────────────┐

│ You’ve completed all available challenges. │

│ Check back tomorrow for new content! │

│ │

│ [ Go to Home ] │

└────────────────────────────────────────────────────┘

**Appendix B: Design Tokens & Style Guide**

**B.1 Color Palette**

* **Modern Beige (Background)**: #F5F1E8
* **Blackout (Primary Text & Buttons)**: #000000
* **Neon Green (Accent)**: #39FF14
* **Soft Pink (Secondary Accent)**: #FFB6C1
* **Dark Gray (Neutral Text)**: #333333
* **Light Gray (Neutral Light)**: #CCCCCC
* **Error Red**: #E53E3E
* **Success Green**: #48BB78
* **Warning Yellow**: #ECC94B

**B.2 Typography**

* **Font Family**: Inter (Google Font)
  + **Weights**: 400 (Regular), 500 (Medium), 600 (Semi-Bold), 700 (Bold)

| **Style** | **Size (sp/pt)** | **Weight** | **Usage** |
| --- | --- | --- | --- |
| Heading 1 | 32sp | Semi-Bold | Main screen titles |
| Heading 2 | 24sp | Semi-Bold | Section/subsection headers |
| Heading 3 | 18sp | Medium | Minor headings |
| Body Text | 16sp | Regular | Paragraphs and descriptions |
| Small Text | 14sp | Regular | Captions, disclaimers, footnotes |
| Button Text | 16sp | Medium | Primary/secondary button labels |

**B.3 Spacing & Layout**

* **Base Unit**: 8px (1 unit)
* **Common Multiples**:
  + 16px (2 units)
  + 24px (3 units)
  + 32px (4 units)
  + 48px (6 units)
* **Container Padding**:
  + Mobile: 16px all around
  + Tablet/Desktop: 32px all around
* **Grid System**:
  + 12-column layout, gutter = 16px (2 × base unit)
* **Corner Radius**:
  + Cards & Modals: 16px (2 × base unit)
  + Buttons: 24px (3 × base unit)
  + Input Fields: 8px (1 × base unit)

**B.4 Iconography**

* **Icon Library**: Lucide (<https://lucide.dev/>)
* **Standard Sizes**:
  + Header/Toolbar Icons: 24 × 24 px
  + Inline/Text Icons: 16 × 16 px
  + Large Action Icons (e.g., thumbs up/down): 48 × 48 px

**B.5 Button Styles**

| **Button Type** | **Background Color** | **Text Color** | **Border Radius** | **Padding** | **Usage** |
| --- | --- | --- | --- | --- | --- |
| Primary Button | #000000 | #FFFFFF | 24px | 16px vertical, 8px horizontal | Main CTAs (e.g., “Continue”) |
| Secondary Button | transparent | #000000 | 24px | 16px vertical, 8px horizontal | Secondary CTAs (e.g., “Play Again”) |
| Link Button | transparent | #39FF14 | — | — | Text links (e.g., “Log In”) |
| Destructive | #E53E3E | #FFFFFF | 24px | 16px vertical, 8px horizontal | Delete, Logout, critical actions |

* **Focus State**: Outline 2px solid #39FF14, no background change.
* **Disabled State**: Opacity 50%, cursor: not-allowed.

**Appendix C: API Endpoints Reference**

Summarizes key REST/GraphQL endpoints including path, method, description, and request/response examples (JSON).

**C.1 Event Ingestion (Layer 1)**

1. **POST** /events/play
   * **Purpose**: Record a “play” event when a user finishes or exits a micro-game.
   * **Request Payload (JSON)**:

json

CopyEdit

{

"user\_id\_hash": "string",

"game\_id": "string",

"session\_id": "string",

"timestamp": "ISO8601 string",

"duration\_sec": integer

}

* + **Response (JSON)**:

json

CopyEdit

{

"status": "success",

"event\_id": "uuid"

}

1. **POST** /events/share
   * **Purpose**: Record when a user shares a micro-game result.
   * **Request Payload**:

json

CopyEdit

{

"user\_id\_hash": "string",

"game\_id": "string",

"share\_channel": "string", // e.g., "twitter", "facebook"

"timestamp": "ISO8601 string"

}

* + **Response**:

json

CopyEdit

{

"status": "success",

"event\_id": "uuid"

}

1. **POST** /events/feedback
   * **Purpose**: Record user feedback (thumbs up/down and optional text).
   * **Request Payload**:

json

CopyEdit

{

"user\_id\_hash": "string",

"game\_id": "string",

"feedback\_label": "up" | "down",

"comment\_text": "string (optional)",

"timestamp": "ISO8601 string"

}

* + **Response**:

json

CopyEdit

{

"status": "success",

"event\_id": "uuid"

}

**C.2 Scoring & Clustering (Layer 2)**

1. **GET** /score/channel/{channel\_id}
   * **Purpose**: Return the current Health Score and metrics for a specific channel.
   * **Response (JSON)**:

json

CopyEdit

{

"channel\_id": "string",

"health\_score": number, // 0 – 100

"virality\_index": number,

"monetization\_index": number,

"sentiment\_ratio": number, // positive / total

"last\_updated": "ISO8601 string"

}

1. **GET** /cluster/brief/{cluster\_id}
   * **Purpose**: Retrieve the Channel Brief JSON for a given cluster.
   * **Response (JSON)**:

json

CopyEdit

{

"channel\_id": "string",

"title": "string",

"description": "string",

"art\_style\_tags": ["string", ...],

"audio\_style\_tags": ["string", ...],

"ui\_template\_id": "string",

"asset\_suggestions": {

"art\_samples": ["url1", "url2"],

"audio\_samples": ["url1", "url2"]

},

"health\_score": number,

"timestamp": "ISO8601 string"

}

1. **POST** /cluster/approve
   * **Purpose**: Curator approves a cluster, triggering a prototype build.
   * **Request Payload**:

json

CopyEdit

{

"cluster\_id": "string",

"approved\_by": "curator\_user\_id",

"timestamp": "ISO8601 string"

}

* + **Response**:

json

CopyEdit

{

"status": "queued",

"build\_id": "string",

"estimated\_time\_sec": integer

}

**C.3 Complex App Services (Layer 3)**

1. **POST** /user/profile
   * **Purpose**: Create or update a user profile (demographics, mood, interests).
   * **Request Payload**:

json

CopyEdit

{

"user\_id\_hash": "string",

"age\_range": "18-24" | "25-34" | "35-44" | …,

"goals": ["string", ...],

"mood\_baseline": integer, // 0 – 10

"interest\_tags": ["string", ...]

}

* + **Response**:

json

CopyEdit

{

"status": "success"

}

1. **GET** /user/dashboard/{user\_id\_hash}
   * **Purpose**: Return personalized dashboard data: daily challenge, progress, community highlights.
   * **Response (JSON)**:

json

CopyEdit

{

"daily\_challenge": {

"id": "string",

"title": "string",

"progress": { "completed\_steps": integer, "total\_steps": integer }

},

"recent\_progress": [

{ "challenge\_id": "string", "completed\_on": "ISO8601 string" },

...

],

"community\_highlights": [

{ "post\_id": "string", "user": "string", "text": "string", "timestamp": "ISO8601 string" },

...

]

}

1. **POST** /chat/message
   * **Purpose**: Send a user message to the AI coach; retrieve a response.
   * **Request Payload**:

json

CopyEdit

{

"user\_id\_hash": "string",

"message": "string",

"context": { "last\_bot\_message\_id": "string (optional)" }

}

* + **Response**:

json

CopyEdit

{

"bot\_message\_id": "string",

"reply": "string",

"suggestions": ["string", ...] // quick-reply buttons

}

**Appendix D: Data Schema & Sample Records**

Defines core tables/collections and example entries. Adjust naming and types for MongoDB, PostgreSQL, DynamoDB, etc.

**D.1 Users Table (MongoDB or PostgreSQL)**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| user\_id\_hash | String (PK) | Hashed UUID to anonymize PII |
| email\_encrypted | String | AES-256 encrypted email address |
| age\_range | String | e.g., “18-24”, “25-34”, “35-44” |
| mood\_baseline | Integer | Last recorded mood (0 – 10) |
| interest\_tags | Array<String> | List of selected interests |
| consent\_analytics | Boolean | Opt-in analytics |
| consent\_personalize | Boolean | Opt-in personalization |
| consent\_marketing | Boolean | Opt-in marketing communications |
| created\_at | DateTime | Account creation timestamp |
| updated\_at | DateTime | Last update timestamp |

**Sample Record:**

json

CopyEdit

{

"user\_id\_hash": "a9f5b6c1-3f72-4d2c-9aef-1234567890ab",

"email\_encrypted": "U2Fsd...encryptedText...==",

"age\_range": "25-34",

"mood\_baseline": 7,

"interest\_tags": ["Daily Affirmations", "Community Support"],

"consent\_analytics": true,

"consent\_personalize": false,

"consent\_marketing": false,

"created\_at": "2025-06-07T02:15:30Z",

"updated\_at": "2025-06-07T02:15:30Z"

}

**D.2 Micro-Game Events Table (Redshift or DynamoDB)**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| event\_id | String (PK) | UUID |
| user\_id\_hash | String | Foreign key to Users |
| game\_id | String | Identifier for micro-game archetype |
| session\_id | String | Unique session ID (UUID) |
| event\_type | String | “play” | “share” | “feedback” |
| timestamp | DateTime | Event timestamp |
| duration\_sec | Integer (opt.) | (for “play” events) length in seconds |
| share\_channel | String (opt.) | (for “share” events) e.g., “twitter” |
| feedback\_label | String (opt.) | (for “feedback”) “up” | “down” |
| comment\_text | String (opt.) | Feedback comment |

**Sample Record:**

json

CopyEdit

{

"event\_id": "f47ac10b-58cc-4372-a567-0e02b2c3d479",

"user\_id\_hash": "a9f5b6c1-3f72-4d2c-9aef-1234567890ab",

"game\_id": "ghost\_tapper\_v1",

"session\_id": "session-abc-123",

"event\_type": "play",

"timestamp": "2025-06-07T02:20:15Z",

"duration\_sec": 45

}

**D.3 Channel Briefs Table (DynamoDB)**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| channel\_id | String (PK) | Unique cluster identifier |
| title | String | Human-readable channel title |
| description | String | Summarized theme |
| art\_style\_tags | Array<String> | e.g., ["neon", "retro"] |
| audio\_style\_tags | Array<String> | e.g., ["synthwave", "ambient"] |
| ui\_template\_id | String | Reference to UI template for prototyping |
| asset\_suggestions | Object | { art\_samples: [url1, url2], audio\_samples: [url1, url2] } |
| health\_score | Number | Latest computed score (0 – 100) |
| latest\_brief\_ts | DateTime | Timestamp of most recent brief |
| status | String | “approved” | “on\_hold” | “rejected” |

**Sample Record:**

json

CopyEdit

{

"channel\_id": "ghost\_drama\_cluster\_20250607",

"title": "Ghost Drama",

"description": "Tap to uncover hidden souls in a neon retro haunted mansion.",

"art\_style\_tags": ["neon", "retro"],

"audio\_style\_tags": ["synthwave"],

"ui\_template\_id": "template\_v2\_neon\_dark",

"asset\_suggestions": {

"art\_samples": ["https://s3.amazonaws.com/appforge-assets/ghost\_sample1.png"],

"audio\_samples": ["https://s3.amazonaws.com/appforge-assets/synthwave\_loop.mp3"]

},

"health\_score": 87,

"latest\_brief\_ts": "2025-06-07T02:45:00Z",

"status": "approved"

}

**D.4 Complex App Challenge Progress (PostgreSQL)**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| progress\_id | String (PK) | UUID |
| user\_id\_hash | String | Foreign key to Users |
| challenge\_id | String | Unique identifier for challenge |
| step\_number | Integer | Current step index (1 – N) |
| completed\_at | DateTime | Timestamp when this step was completed |
| data\_payload | JSON | Additional data (e.g., answers, journal text) |

**Sample Record:**

json

CopyEdit

{

"progress\_id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",

"user\_id\_hash": "a9f5b6c1-3f72-4d2c-9aef-1234567890ab",

"challenge\_id": "breakup\_challenge\_find\_closure",

"step\_number": 2,

"completed\_at": "2025-06-07T03:10:00Z",

"data\_payload": {

"chosen\_strategy": "Write gratitude list",

"journal\_entry": "I am grateful for supportive friends, good health, and fresh coffee each morning."

}

}

**Appendix E: Glossary of Key Terms**

Quick definitions of frequently used terminology throughout the documentation.

* **Channel Brief**  
  A JSON document summarizing a cluster of related micro-games. Contains title, description, asset suggestions, health metrics, and metadata.
* **Cluster/Channel**  
  A grouping of micro-games that share similar themes or user engagement patterns. Formed via ML clustering.
* **Health Score**  
  A composite metric (0 – 100) measuring a channel’s viability, derived from virality, monetization, and sentiment indices.
* **Lifecycle of an Asset**  
  The progression of an AI-generated art or audio file—from creation, tagging, distribution in micro-games, archiving, to deletion after 30 days of inactivity.
* **Micro-Game Archetype**  
  A predefined template (e.g., Tapper, Quiz, Idle, Simulator) used to spin up lightweight, testable experiences quickly.
* **Pipeline**  
  The end-to-end flow:
  1. **Layer 1 (Micro-Games)**: User plays small experiments.
  2. **Layer 2 (Curation)**: Data is aggregated, scored, and clustered.
  3. **Layer 3 (Complex Apps)**: High-potential themes are built into fully featured applications.
* **Trend Token API**  
  A paid endpoint that exposes real-time channel metrics and cluster briefs to third-party developers.
* **Right-to-Know / Right-to-Erase**  
  Privacy features allowing users to request an export or deletion of their data to comply with regulations (GDPR, CCPA).
* **Self-Harm Escalation**  
  Automated moderation logic that, upon detecting self-harm or violent content, alerts Trust & Safety and sends supportive resources to the user.
* **UX Flow**  
  The sequence of screens and interactions a user follows to complete a task—for example, signing up or playing a micro-game.

**Appendix F: Security & Compliance Checklist**

High-level checklist to ensure the platform meets security, privacy, and compliance requirements.

**F.1 Data Encryption**

* TLS 1.3 for all API endpoints (API Gateway, web clients, load balancers).
* AES-256 encryption at rest for S3 buckets, RDS, DynamoDB, and Elasticache.
* mTLS (mutual TLS) for service-to-service communication within the Kubernetes cluster.

**F.2 Authentication & Authorization**

* OAuth 2.0 via Auth0 (or equivalent) for user login.
* Role-Based Access Control (RBAC) in mobile/web apps for admin, curator, and user roles.
* AWS IAM policies with least-privilege for Lambda functions, ECS tasks, EC2 instances.

**F.3 Privacy & Consent**

* Consent toggles enforced on event ingestion (e.g., Analytics & Performance cannot be disabled if user continues).
* Consent logs stored with timestamp in DynamoDB.
* Right-to-Erase endpoint deletes user data from MongoDB, DynamoDB, S3, and Redshift.
* Right-to-Know endpoint exports user data from all relevant stores into a ZIP and emails.

**F.4 Vulnerability Management**

* Weekly Snyk or similar scans for open-source dependencies.
* Quarterly penetration tests by an external security firm.
* Docker images scanned by Clair or Aqua before production deployment.

**F.5 Monitoring & Alerting**

* Prometheus alerts for Kinesis backlog > 10,000 records.
* Grafana dashboards for CPU, memory, latency, GPT API call duration.
* Sentry alerts for unhandled exceptions with priority ≥ P2.

**F.6 Content Moderation**

* AI moderation pipeline (AWS Rekognition for images/videos; Perspective API for text).
* Human-in-the-Loop dashboard for reviewing flagged content.
* Escalation workflow: self-harm or violent content triggers PagerDuty alert and a supportive push notification.

**F.7 Disaster Recovery & Backups**

* RDS automated snapshots daily; retained for 30 days.
* MongoDB Atlas point-in-time backups every 6 hours; retained 30 days.
* DynamoDB global tables with on-demand backups.
* S3 bucket lifecycle: transition raw events to Glacier after 90 days.
* Cross-region replication for Redis clusters and DynamoDB as needed.
* Disaster Recovery (DR) drills scheduled every 6 months.

**F.8 Compliance Documentation**

* GDPR: Privacy Policy published; cookie banner implemented.
* CCPA: “Do Not Sell My Data” link present on landing page.
* SOC 2 Type II readiness: Audit logs retention, periodic access reviews.
* Consent records and deletion logs archived for compliance audits.
* Documented data retention policy:
  + Raw events: 180 days.
  + Analytics aggregates: 1 year.
  + User profiles: until account deletion.

**Appendix G: Infrastructure Diagram (Text-Based)**

A high-level overview of each layer’s core components and how they connect. Useful to translate into a visual diagram later.

less

CopyEdit

+----------------------------------------------------------+

| AppForge.AI |

| |

| Layer 1: Micro-Games Pump-and-Dump |

| ┌──────────────────────────┐ ┌──────────────────────┐ |

| │ User Devices (iOS/Android) │ │ Game Clients (Unity, │ |

| │ - Analytics SDK │ │ Flutter, React Native)│ |

| └──────────┬─────────────────┘ └─────────┬────────────┘ |

| │ │ |

| ▼ ▼ |

| [ Event Ingestion API Gateway ] [ Feedback API Gateway ] |

| │ │ |

| ▼ ▼ |

| [ Lambda Preprocessor ] [ Lambda Preprocessor ] |

| │ │ |

| ▼ ▼ |

| [ Kinesis Data Streams ] [ Kinesis Data Streams ] |

| │ │ |

| └──────────┬────────────────────────┘ |

| ▼ |

| [ Real-Time Analytics (Kinesis Analytics) ] |

| │ |

| ▼ |

| [ DynamoDB / MongoDB ] |

| |

| Layer 2: Trend Aggregator & Channel Curator |

| ┌──────────────────────────┐ ┌──────────────────────┐ |

| │ Scoring Microservice (Node.js) │ │ Embedding Service (Python + GPU) │ |

| └──────────┬─────────────────┘ └─────────┬────────────┘ |

| │ │ |

| ▼ ▼ |

| [ RabbitMQ / SQS / Topics ] [ SageMaker Endpoint ] |

| │ │ |

| [ BERT Sentiment Model ] [ Sentence-BERT ] |

| │ │ |

| └─────────┬───────────────────────┘ |

| ▼ |

| [ UMAP (2D/3D Reduction) ] |

| │ |

| ▼ |

| [ HDBSCAN Clustering → Channel Brief JSON ] |

| │ |

| ▼ |

| [ S3 (Channel Briefs) ] [ DynamoDB (Index) ] |

| │ |

| ▼ |

| [ Curator Dashboard (React + AppSync + DynamoDB) ] |

| |

| Layer 3: Complex App Forge |

| ┌──────────────────────────┐ ┌──────────────────────┐ |

| │ User Profile Service (Django + PostgreSQL) │ │ GPT Content Service (Flask + Redis Cache) │ |

| └──────────┬─────────────────┘ └─────────┬────────────┘ |

| │ │ |

| ▼ ▼ |

| [ SDXL Style Service (EC2 GPU) ] [ ElevenLabs Audio ] |

| │ │ |

| [ Style Assets (S3) ] [ Audio Assets (S3) ] |

| │ │ |

| └──────────┬────────────────────────┘ |

| ▼ |

| [ MongoDB Atlas (User Data, Posts) ] |

| │ |

| [ DynamoDB (Channel Metadata) ] |

| │ |

| [ Real-Time (AppSync, Redis Pub/Sub) ] |

| │ |

| [ Frontend Clients (React Native, iOS, Android) ] |

| |

| Infrastructure & DevOps |

| - Kubernetes Cluster (EKS) with Namespaces: layer1, layer2, layer3 |

| - CI/CD: GitHub Actions → Argo CD / Jenkins → ECR |

| - Monitoring: Prometheus + Grafana, Sentry, Datadog |

| - Storage: S3, Glacier for archives |

| - Security: AWS WAF, IAM policies, mTLS for services |

| - Disaster Recovery: Multi-region failover, daily snapshots |

+----------------------------------------------------------+

**Appendix H: Roadmap & Milestones**

A timeline of key deliverables and success metrics for the first 12 months.

**H.1 Q1 (Month 1–3)**

**Deliverable:** MVP Micro-Game Suite, Basic Pipeline

* **Deploy** 5 micro-game archetypes (Tapper, Quiz, Idle, Simulator, Text-Drama).
* **Implement** event ingestion (Kinesis) and real-time analytics (Kinesis Analytics).
* **Launch** initial 5 titles: “Ghost Tapper,” “Quick Quiz,” “Idle Spinner,” “Ghost Threads,” “Text Drama V1.”
* **Build** privacy/consent flow, onboarding overlay.
* **Release** Curator Dashboard (basic Channel Overview).

**KPIs:**

* 10,000 installs across two micro-games.
* 15% feedback opt-in rate.
* 100 clusters formed, 20 Channel Briefs published.

**H.2 Q2 (Month 4–6)**

**Deliverable:** Pipeline Refinement & First Complex App

* **Enhance** AI sentiment model, deploy BERT in SageMaker.
* **Implement** UMAP + HDBSCAN clustering.
* **Add** React-based Curator Dashboard with Brief Editor.
* **Prototype** first complex app: “Breakup Bootcamp” (5-step challenge, chat coach).
* **Launch** Beta via TestFlight / Play Beta.

**KPIs:**

* 50 Channel Briefs generated.
* 10 brief approvals → 5 prototypes built.
* Breakup Bootcamp Day-7 retention ≥ 20%.
* $10,000 revenue through micro-games (ads + IAP).

**H.3 Q3 (Month 7–9)**

**Deliverable:** Community & Growth Features

* **Open** API for Trend Token (paid tier).
* **Implement** in-app community feeds, basic moderation (Perspective API).
* **Release** “Loopr Pro” with habit tracker and SWOT analysis feature.
* **Implement** subscription billing ($4.99/month premium tier).
* **Localize** top-performing micro-games to Spanish and Japanese.

**KPIs:**

* 20,000 monthly active users (MAU) across all products.
* $50,000 monthly revenue (IAP + subs + API).
* 50% Month-over-Month (MoM) growth in UAU (unique active users).
* Community feed usage: 5,000 posts, 20,000 comments.

**H.4 Q4 (Month 10–12)**

**Deliverable:** Diversification & Scale

* **Prototype** AR/Voice skill experiences (e.g., “Ghost Filter AR Challenge,” Alexa Skill).
* **Integrate** blockchain loyalty tokens (ERC-20 testnet) for rewards.
* **White-Label** Trend Analytics SDK for at least one enterprise client.
* **Launch** Developer Program: tutorials, SDK docs (GitHub repo).
* **Establish** on-device inference for micro-LLMs (e.g., local personalization).

**KPIs:**

* 100,000 MAU.
* $100,000 monthly revenue.
* 10 external SDK integrations/licenses sold.
* 5% of active installs use blockchain rewards.
* Achieve SOC 2 Type II audit readiness.

**Appendix I: Governance & Ethics Framework**

Documents the high-level policies and procedures for safe, compliant AI usage.

**I.1 Ethics Committee Roles & Responsibilities**

* **Chair (CTO or Senior Engineer)**
  + Oversees quarterly AI model bias audits.
  + Approves new AI features in pipeline (e.g., sensitive content classification).
* **Privacy Officer**
  + Ensures compliance with GDPR/CCPA.
  + Reviews privacy policy updates and consent flows.
* **Trust & Safety Lead**
  + Monitors moderation queue, escalates self-harm or violence content.
  + Coordinates with external mental health resources.
* **Legal & Compliance Advisor**
  + Reviews Terms of Service, End User License Agreements (EULAs).
  + Advises on data retention and regulatory filings.

**I.2 Quarterly AI Bias Audit Checklist**

* Review training data sources for demographic representation (e.g., gender, age, region).
* Run fairness metrics on classifier outputs (e.g., false positive/negative rates by subgroup).
* Document any observed bias; retrain or adjust model weights as needed.
* Publish audit report summary internally (no PII).
* Confirm model versioning and rollback plan in case of high-risk findings.

**I.3 Content Moderation Workflow**

1. **Automated Pre-Filter**
   * All UGC (text/images/video) passes through AWS Rekognition and Perspective API.
   * If flagged (hate, violence, self-harm), it’s routed to human review.
2. **Human Review Dashboard**
   * Moderators see flagged items in priority queue.
   * “Approve” → content goes live.
   * “Reject” → content is deleted; user notified.
   * “Escalate” → severe cases (self-harm, violence) routed to Trust & Safety team.
3. **Escalation Protocol**
   * Immediate PagerDuty alert to Trust & Safety.
   * If self-harm, send push notification to user with mental health resources.
   * Log actions with audit trail (who reviewed, timestamp, decision).

**I.4 Privacy Policy Summary (User-Facing)**

A concise version to present at onboarding under “Learn More.”

1. **Data Collected**
   * Gameplay events (anonymous).
   * Device & performance metrics.
   * User feedback text.
2. **Purpose**
   * Improve game balance, user experience.
   * Detect abusive content.
   * Provide personalized content.
3. **User Rights**
   * **Opt-Out**: Manage consent settings in app under Settings → Privacy.
   * **Right-to-Know**: Request a copy of all collected data via Privacy page.
   * **Right-to-Erase**: Permanently delete account and all data via Privacy page.
4. **Data Retention**
   * Raw events stored 180 days.
   * Aggregated analytics stored 1 year.
   * User account data retained until deletion request.

**Appendix J: Project Risks & Mitigations**

High-level list of key risks identified, with mitigation strategies.

**J.1 Data Privacy & Compliance Risk**

* **Risk**
  + Failure to properly obtain or store consent (GDPR/CCPA violations).
* **Mitigation**
  + Enforce mandatory toggle for analytics during onboarding.
  + Store consent logs with timestamp.
  + Test Right-to-Erase flows each release cycle.

**J.2 Trend Volatility & Irrelevance**

* **Risk**
  + Investing in micro-games that do not generate meaningful data, wasting resources.
* **Mitigation**
  + Implement automatic “sunsetting” for micro-games with < 100 plays in first 7 days.
  + Triangulate signals from at least 3 sources (game feedback, social shares, session length).
  + Curator Dashboard flags low-activity clusters for manual review or deletion.

**J.3 AI Model Bias & Content Safety**

* **Risk**
  + Sentiment model misclassifies feedback, leading to skewed cluster formation.
  + GPT content generation inadvertently produces harmful or biased text.
* **Mitigation**
  + Quarterly bias audits (Appendix I.2).
  + Human-in-the-loop review for top 10% of generated briefs.
  + Safety guardrails built into GPT prompts: explicit instructions to avoid sensitive topics.
  + Continuous retraining of BERT sentiment model with diverse, labeled data.

**J.4 Technical Debt & Scalability**

* **Risk**
  + Rapid prototyping leads to messy code, causing outages under load.
* **Mitigation**
  + Enforce code reviews and linting (ESLint, SonarQube).
  + Adopt microservice architecture with clear service boundaries.
  + Horizontal Pod Autoscaler (HPA) for critical components (Kinesis consumers, API gateways).
  + Scheduled load tests monthly to identify bottlenecks early.

**J.5 User Adoption & Retention**

* **Risk**
  + Users try micro-games but don’t convert to complex apps.
* **Mitigation**
  + Embed subtle calls-to-action in micro-games: “Join the full experience in Breakup Bootcamp.”
  + Use push notifications and email drip to re-engage first-time players.
  + Offer limited-time in-game rewards for downloading the complex app.

**Appendix K: Team & Roles**

Outline of recommended roles for a core AppForge.AI team and suggested responsibilities.

**K.1 Core Team Roles**

1. **Project Lead / Product Manager**
   * Coordinates roadmap, prioritizes features, liaises with stakeholders.
   * Ensures alignment between layers (micro-games, curation, complex apps).
2. **Technical Lead / Architect**
   * Designs system architecture (Kubernetes clusters, data pipeline, AI/ML).
   * Reviews code for consistency, security, and scalability.
3. **Backend Engineers (2–3)**
   * Build event ingestion APIs, scoring microservice, cluster composer.
   * Implement data storage (DynamoDB, MongoDB, PostgreSQL).
   * Setup CI/CD pipelines and infrastructure as code (Terraform, Helm charts).
4. **Frontend Engineers (2)**
   * Build mobile clients (React Native or native iOS/Android).
   * Develop Curator Dashboard (React + AppSync).
   * Ensure UI matches design tokens (Appendix B).
5. **Data Engineer / ML Engineer**
   * Develop sentiment analysis pipeline (BERT, SageMaker).
   * Implement UMAP + HDBSCAN clustering jobs.
   * Maintain data warehouse (Redshift, Airflow ETLs).
6. **DevOps / SRE Engineer**
   * Manage Kubernetes (EKS), monitoring (Prometheus, Grafana).
   * Oversee security configurations (IAM roles, WAF, backups).
   * Run DR drills and ensure high availability.
7. **AI Prompt Engineer / Designer**
   * Craft GPT-4 prompts for brief summarization, content generation.
   * Maintain asset taxonomy and style catalogs.
   * Iterate on SDXL style transfer pipelines.
8. **UI/UX Designer**
   * Create high-fidelity mockups from wireframes (Appendix A).
   * Define style guides, iconography, spacing (Appendix B).
   * Conduct usability testing for micro-games and complex apps.
9. **Trust & Safety / Moderator Lead**
   * Oversee content moderation (automated + human).
   * Manage escalation protocols (self-harm, violence).
   * Coordinate with mental health partners for user support.
10. **Legal & Compliance Officer**
    * Draft Terms of Service, Privacy Policy.
    * Ensure platform compliance with GDPR/CCPA, SOC 2.
    * Conduct regular audits (Appendix F).

**Appendix L: Sample User Journey Narratives**

Illustrative “day in the life” narratives to show how different user personas interact with the system.

**L.1 Persona: Jasmine, 26, Seeking Emotional Support**

1. **Discovery**
   * Jasmine sees an Instagram ad for “Breakup Bootcamp: 5 Days to Heal” and taps.
   * She is prompted to download the app and signs up via email.
2. **Onboarding**
   * Jasmine enters her email/password, completes the Privacy & Consent toggles (accepts Analytics, declines Marketing).
   * She sees the onboarding slider: sets mood baseline to 4, selects goals (“Find Closure,” “Community Support”).
   * She finishes and enters the app.
3. **First Micro-Game**
   * Jasmine is shown “Ghost Threads Challenge,” a Tapper game.
   * She taps for 30 seconds, watches an ad to get bonus time, then fills out brief feedback.
4. **Curated Brief**
   * A day later, Jasmine receives a push notification: “New theme ready: Ghost Drama.”
   * She downloads “Ghost Drama” micro-game and plays again.
5. **Complex App Invitation**
   * After playing two micro-games, Jasmine gets an in-app prompt:  
     “Ready to level up? Try Breakup Bootcamp for a deeper journey.”
   * She taps “Learn More” and sees a short preview video.
6. **Breakup Bootcamp**
   * Jasmine enters the Beta for “Breakup Bootcamp.”
   * She completes steps: writes in a journal, chooses coping strategies, interacts with AI Coach.
   * She visits the community feed to read other users’ experiences.
7. **Retention & Growth**
   * Over five days, she completes all steps and shares progress on social media.
   * The AI Coach invites her to subscribe ($4.99/month) for premium content (guided meditations, live group sessions).
   * Jasmine opts in, feeling more supported.

**L.2 Persona: Carlos, 30, Indie Game Developer**

1. **Onboarding as Curator**
   * Carlos registers as a Curator using his work email.
   * He logs into Curator Dashboard, sees a list of new clusters.
2. **Reviewing Briefs**
   * Carlos filters to show channels with Health Score > 80.
   * He clicks “Ghost Drama Cluster,” reviews metrics, user feedback, asset suggestions.
   * He tweaks the JSON: adjusts description, adds a new UI template, then clicks “Publish & Build Prototype.”
3. **Monitoring Builds**
   * Carlos receives a Slack webhook: “Prototype for Ghost Drama ready.”
   * He clicks the link, tests the new micro-game build on staging device.
   * He notes a bug: ad doesn’t load on Android 11. He files a ticket back in the backlog.
4. **Pipeline Feedback**
   * Carlos writes a brief note in the Dashboard: “Consider adding a ‘dark mode’ for this cluster’s micro-games.”
   * He assigns a task to the Mobile team to add a theme toggle in version v1.1.

**Appendix M: Testing & QA Checklists**

Outlines functional, integration, and performance tests to validate each layer.

**M.1 Functional Testing (End-User)**

1. **Sign-Up Flow**
   * Verify “Sign Up” displays correctly.
   * Test valid/invalid email & password.
   * Ensure “Continue” disabled until valid.
   * Confirm privacy toggles enforce at least Analytics.
   * Verify “Learn More” opens policy modal.
   * Test “I Understand” closes modal, returns to toggles.
   * Confirm “Accept & Continue” → confirmation screen.
2. **Micro-Game Play**
   * Launch each archetype (Tapper, Quiz, Idle, Simulator, Text Drama).
   * Simulate game play: confirm events (play, share, feedback) sent to backend.
   * Test rewarded ad flow: “Watch Ad” triggers correct callback.
   * Check feedback modal functionality (submit vs skip).
3. **Curator Dashboard**
   * Login with valid/invalid credentials.
   * Confirm channel list updates with sample data.
   * Test filtering, sorting, pagination.
   * Approve/Reject channel → ensure correct API call.
   * Edit JSON brief → “Save Changes” updates backend.
   * “Publish & Build Prototype” triggers build and notifies curator.
4. **Complex App**
   * On first launch, test onboarding screens (persona selection, mood, interests).
   * Verify daily challenge card displays correct data.
   * Complete a challenge step → data stored in DB.
   * Test AI Chat Coach: send sample message, receive response.
   * Post to community feed → visible to other test users.
   * Subscription purchase flow (sandbox mode).

**M.2 Integration Testing (Backend Services)**

1. **Event Ingestion to Clustering**
   * Simulate batch of “play” and “feedback” events → verify Kinesis streams.
   * Run Kinesis Data Analytics job → confirm aggregated metrics stored in DynamoDB.
   * Trigger BERT sentiment service → confirm sentiment stored correctly.
   * Run UMAP + HDBSCAN clustering script → ensure clusters created in S3.
2. **Curator → Build Trigger**
   * Approve cluster via API → verify EventBridge message.
   * Confirm build pipeline receives event, spins up Kubernetes job.
   * Mock build success → ensure new micro-game binary appears in ECR.
   * Curator receives Slack webhook confirming build completion.
3. **Complex App Data Flow**
   * Profile update → verify PostgreSQL record.
   * GPT content generation request → test caching via Redis.
   * SDXL style transfer job → ensure generated assets appear in S3.
   * Check dynamic audio mixer → confirm correct audio served via CDN.

**M.3 Performance & Load Testing**

1. **Kinesis Streams**
   * Simulate 10,000 events/sec → confirm no data loss.
   * Monitor shard scaling: ensure autoscaler adjusts shard count.
2. **Clustering Jobs**
   * Test embedding generation for 10,000 comments → measure latency.
   * Run UMAP + HDBSCAN on 1 million embeddings → record CPU/GPU usage.
3. **API Gateways**
   * Stress test /events/play with 50 requests/sec → monitor 95th percentile latency < 100ms.
   * Test /cluster/approve endpoint under load (100 concurrent requests).
4. **Complex App Backend**
   * Simulate 5,000 concurrent chat requests to AI Coach → ensure SageMaker endpoints autoscale.
   * Load test user dashboard API with 2,000 concurrent users.

**Appendix N: Release & Versioning Strategy**

**N.1 Semantic Versioning**

* **Scheme**: MAJOR.MINOR.PATCH
  + **MAJOR**: Incompatible API changes.
  + **MINOR**: Backward-compatible new features.
  + **PATCH**: Bug fixes and minor improvements.

**N.2 Branching Strategy (GitHub)**

* **Main Branch (main)**: Always production-ready.
* **Development Branch (develop)**: Integrates features for next minor release.
* **Feature Branches (feature/\*)**: Named after Jira ticket or feature (e.g., feature/cluster-visual).
* **Release Branches (release/vX.Y)**: Created from develop when preparing a new minor release.
* **Hotfix Branches (hotfix/X.Y.Z)**: Created from main when a critical bug is found in production.

**N.3 CI/CD Pipeline (GitHub Actions → Argo CD)**

1. **Feature Branch PR**
   * Runs linting (ESLint, SonarQube), unit tests (Jest, PyTest, JUnit, Unity).
   * On PR approval, merges into develop.
2. **Develop Branch Merge**
   * Triggers integration tests (Docker build, Kubernetes helm chart validation).
   * If successful, automatically deploys to staging namespace in EKS.
3. **Release Branch Creation**
   * Manual trigger: run “release pipeline.”
   * Generate changelog via semantic-release.
   * Tag new version (e.g., v1.2.0).
   * Publish Docker images to ECR and Helm charts to Nexus.
   * Argo CD deploys to production namespace after manual approval.
4. **Hotfix Merge**
   * Merges into main and develop.
   * Patch version bumped automatically.
   * Deploys to production immediately.

**Appendix O: Partner & Licensing Model**

**O.1 Data-as-a-Service (Trend Token API)**

* **Free Tier**:
  + Up to 100,000 calls/month.
  + Rate limit: 10 calls/sec.
* **Paid Tier**:
  + $0.01 per call after free tier.
  + Rate limit: 100 calls/sec.
* **Enterprise Tier**:
  + Custom pricing for > 10 million calls/month.
  + Includes dedicated SLA and priority support.

**O.2 SDK Licensing**

* **Trend Analytics SDK**
  + Provides embeddings, clustering insights, and health scores for a given dataset.
  + License tiers:
    1. **Basic**: $10,000 one-time fee; includes annual updates.
    2. **Pro**: $25,000 one-time; includes priority support and custom feature requests.
    3. **Enterprise**: $50,000+ one-time; includes on-premise deployment, white-label components.
* **Usage Terms**
  + License covers up to 3 developer seats, 2 environments (staging, production).
  + Unlimited API calls within developer’s own cloud account.
  + Annual renewal required for Pro and Enterprise to receive updates.

**Appendix P: Communication & Support Channels**

**P.1 Internal Communication**

* **Slack Channels**
  + #dev-backend – Backend engineering discussions.
  + #dev-frontend – Frontend and mobile engineering.
  + #ml-team – Data science and ML topics.
  + #qa – Status of ongoing tests and bug reports.
  + #ops-alerts – Prometheus alerts and SRE coordination.
  + #curator-notifications – Automated Slack webhooks for new cluster briefs.
* **Email**
  + dev-team@appforge.ai – General development coordination.
  + safety@appforge.ai – Trust & Safety issues.
  + legal@appforge.ai – Compliance/legal questions.

**P.2 External Support & Documentation**

* **Public Documentation** (GitHub Pages)
  + API references, SDK guides, quickstarts, and FAQ.
  + URL: https://docs.appforge.ai/
* **Developer Forum** (Discourse)
  + Category: appforge-developers for integration questions, best practices.
* **Issue Tracking** (Jira)
  + Projects: APF-Backend, APF-Frontend, APF-ML, APF-QA.