

# Persona Schema - Pen Name Character Definition System

## Core Persona Structure

### Persona Entity Definition

```
interface PenNamePersona {  
    // Identity  
    id: string; // UUID  
    penName: string; // Public pen name (e.g., "Maria Santos")  
    status: PersonaStatus;  
  
    // Character Profile  
    characterDemographics: CharacterDemographics;  
    personalityTraits: PersonalityProfile;  
    backstory: string;  
    writingThemes: ContentThemes;  
  
    // Geographic Setting  
    fictionalLocation: GeographicProfile;  
  
    // Content Strategy  
    contentStrategy: ContentStrategy;  
    platformPersonas: PlatformPersonas;  
  
    // Technical Configuration  
    proxyConfiguration: ProxyAssignment;  
  
    // Performance Tracking  
    metrics: PersonaMetrics;  
  
    // Metadata  
    createdAt: Date;  
    updatedAt: Date;  
    lastUsed: Date;  
}
```

## Persona Status Types

```
type PersonaStatus =  
    | 'active'          // Currently posting content  
    | 'inactive'       // Created but not currently used  
    | 'archived'       // Retired persona, kept for reference  
    | 'developing'     // Being refined before activation  
    | 'suspended';    // Temporarily disabled due to issues
```

## Character Demographics

### Demographic Profile Structure

```
interface CharacterDemographics {  
    // Basic Character Info  
    age: number; // Fictional age (impacts content style)  
    gender: 'male' | 'female' | 'non-binary' | 'prefer-not-to-say';  
    occupation: string; // Fictional job (teacher, artist, entrepreneur)  
    educationLevel: EducationLevel;  
  
    // Social Background  
    socioeconomicStatus: 'working-class' | 'middle-class' |  
    'upper-middle' | 'affluent';  
    familyStatus: 'single' | 'married' | 'divorced' |  
    'widowed' | 'complicated';  
    hasChildren: boolean;  
  
    // Cultural Identity  
    culturalBackground: string[]; // ['Filipino', 'Chinese', 'Spanish']  
    languages: LanguageProfile[];  
    religion?: string; // Optional, influences content themes  
  
    // Lifestyle Indicators  
    urbanRural: 'urban' | 'suburban' | 'rural';  
    lifestyle: LifestyleProfile;  
}  
  
interface EducationLevel {
```

```

    level: 'high-school' | 'college' | 'graduate' |
'professional';
    field?: string; // Engineering, Arts, Business, etc.
    institution?: string; // For authenticity in content
}

interface LanguageProfile {
    language: 'english' | 'filipino' | 'tagalog' | 'cebuan'
| 'hiligaynon';
    proficiency: 'native' | 'fluent' | 'conversational' |
'basic';
    useInContent: boolean; // Whether to post in this
language
}

interface LifestyleProfile {
    interests: string[]; // ['reading', 'travel', 'cooking',
'technology']
    hobbies: string[]; // ['photography', 'gardening',
'music']
    values: string[]; // ['family', 'success', 'creativity',
'spirituality']
    personalityType?: string; // MBTI, Enneagram, etc.
}

```

## Personality Profile System

### Core Personality Traits

```

interface PersonalityProfile {
    // Communication Style
    communicationStyle: CommunicationStyle;

    // Content Preferences
    contentPreferences: ContentPreferences;

    // Social Behavior
    socialBehavior: SocialBehavior;

    // Emotional Expression
    emotionalExpression: EmotionalProfile;
}

```

```

    // Intellectual Approach
    intellectualStyle: IntellectualStyle;
}

interface CommunicationStyle {
    formality: 'very-formal' | 'formal' | 'casual' | 'very-casual' | 'slang';
    verbosity: 'concise' | 'moderate' | 'verbose';
    humor: 'serious' | 'dry' | 'playful' | 'sarcastic' | 'wholesome';
    emoticons: 'never' | 'rarely' | 'sometimes' | 'frequently';
    punctuation: 'minimal' | 'standard' | 'excessive'; // affects ... usage, !!!
    typos: 'perfect' | 'occasional' | 'frequent'; // authentic imperfection
}

interface ContentPreferences {
    preferredTopics: string[]; // Primary content themes
    avoidedTopics: string[]; // Topics this persona never discusses
    contentMix: {
        personal: number; // % personal stories/experiences
        educational: number; // % informative/helpful content
        entertainment: number; // % humor/fun content
        promotional: number; // % business/product content
        social: number; // % community/social issues
    };
    hashtagStyle: 'minimal' | 'moderate' | 'heavy' | 'trending-focused';
}

interface SocialBehavior {
    engagementLevel: 'lurker' | 'selective' | 'active' | 'very-active';
    responseTime: 'immediate' | 'quick' | 'delayed' | 'sporadic';
}

```

```
    controversyTolerance: 'avoids' | 'neutral' | 'engages' |  
'seeks';  
    supportiveness: 'critical' | 'neutral' | 'encouraging' |  
'cheerleader';  
    networking: 'private' | 'selective' | 'open' |  
'aggressive';  
}
```

## Geographic Profile

### Location-Based Character Development

```
interface GeographicProfile {  
    // Primary Location  
    primaryLocation: PhilippinesLocation;  
    timezone: 'Asia/Manila';  
  
    // Location History  
    previousLocations?: LocationHistory[];  
    travelExperience: TravelProfile;  
  
    // Local Context  
    localKnowledge: LocalKnowledge;  
    regionalDialect?: RegionalDialect;  
}  
  
interface PhilippinesLocation {  
    region: 'NCR' | 'CAR' | 'Region I' | 'Region II' |  
'Region III' | 'Region IV-A' |  
        'Region IV-B' | 'Region V' | 'Region VI' | 'Region  
VII' | 'Region VIII' |  
        'Region IX' | 'Region X' | 'Region XI' | 'Region  
XII' | 'Region XIII' | 'BARMM';  
    province: string;  
    city: string; // Manila, Cebu City, Davao, etc.  
    district?: string; // For Metro Manila (Makati, BGC,  
etc.)  
    neighborhood?: string; // Specific area knowledge  
}  
  
interface LocalKnowledge {
```

```

    landmarks: string[]; // Known local places for authentic
references
    transportation: string[]; // MRT, jeepney routes, etc.
    foodPlaces: string[]; // Local restaurants, street food
    events: string[]; // Local festivals, events
    slang: string[]; // Regional expressions
    culturalNuances: string[]; // Local customs, behaviors
}

```

```

interface TravelProfile {
    domesticTravel: string[]; // Other Philippine locations
visited
    internationalTravel: string[]; // Countries visited
(affects worldview)
    travelFrequency: 'never' | 'rare' | 'occasional' |
'frequent';
    travelStyle: 'budget' | 'mid-range' | 'luxury' |
'backpacker' | 'business';
}

```

## **Content Strategy Framework**

### **Strategic Content Planning**

```

interface ContentStrategy {
    // Publishing Schedule
    postingSchedule: PostingSchedule;

    // Content Distribution
    contentMix: ContentMixStrategy;

    // Engagement Strategy
    engagementStrategy: EngagementStrategy;

    // Growth Strategy
    growthStrategy: GrowthStrategy;

    // Risk Management
    riskManagement: RiskManagementStrategy;
}

```

```

interface PostingSchedule {
    frequency: {
        daily: number; // Posts per day
        weekly: number; // Total posts per week
        monthly: number; // Target monthly posts
    };

    timeSlots: {
        morning: TimeSlot; // 6AM-12PM
        afternoon: TimeSlot; // 12PM-6PM
        evening: TimeSlot; // 6PM-12AM
        lateNight: TimeSlot; // 12AM-6AM
    };

    weekdayPattern: {
        monday: DaySchedule;
        tuesday: DaySchedule;
        wednesday: DaySchedule;
        thursday: DaySchedule;
        friday: DaySchedule;
        saturday: DaySchedule;
        sunday: DaySchedule;
    };

    seasonalAdjustments: SeasonalPattern[];
}

interface TimeSlot {
    probability: number; // 0-1, likelihood of posting in
    this slot
    contentTypes: string[]; // Preferred content types for
    this time
    engagement: 'low' | 'medium' | 'high'; // Expected
    engagement level
}

interface ContentMixStrategy {
    contentTypes: {
        textOnly: number; // % text-only posts
        imageWithText: number; // % image + caption posts
    }
}

```

```

        videoContent: number; // % video posts
        linkSharing: number; // % shared articles/links
        polls: number; // % interactive polls
        stories: number; // % story content (Instagram/
Facebook)
    };

    themes: {
        personal: ThemeStrategy;
        professional: ThemeStrategy;
        entertainment: ThemeStrategy;
        educational: ThemeStrategy;
        social: ThemeStrategy;
    };
}

interface ThemeStrategy {
    frequency: number; // % of total content
    preferredDays: string[]; // ['monday', 'wednesday']
    preferredTimes: string[]; // ['morning', 'evening']
    engagement: EngagementExpectation;
}

```

## **Platform-Specific Personas**

### **Multi-Platform Character Consistency**

```

interface PlatformPersonas {
    twitter: TwitterPersona;
    instagram: InstagramPersona;
    tiktok: TikTokPersona;
    linkedin: LinkedInPersona;
    facebook: FacebookPersona;
}

interface TwitterPersona {
    // Twitter-specific traits
    tweetStyle: 'threads' | 'single-tweets' | 'retweets' |
'mixed';
    hashtagUsage: 'minimal' | 'strategic' | 'trending' |
'excessive';
}

```



```

    engagementStyle: 'lurker' | 'replier' | 'quoter' |
'original';

    // Content adaptation
    characterLimit: 'concise' | 'thread-master' | 'link-
heavy';
    replyBehavior: 'supportive' | 'questioning' | 'witty' |
'educational';

    // Twitter-specific topics
    twitterTopics: string[]; // Topics particularly suited
for Twitter
    avoidedOnTwitter: string[]; // Topics to avoid on this
platform

    // Engagement patterns
    followingStrategy: 'selective' | 'growth-focused' |
'quality-over-quantity';
    retweetStrategy: 'rare' | 'strategic' | 'frequent' |
'curator';
}

interface InstagramPersona {
    // Visual content strategy
    aestheticStyle: 'minimalist' | 'colorful' | 'dark' |
'bright' | 'vintage' | 'modern';
    photoStyle: 'professional' | 'candid' | 'artistic' |
'lifestyle' | 'documentary';

    // Instagram-specific features
    storiesUsage: 'daily' | 'occasional' | 'rare' | 'event-
only';
    reelsStrategy: 'trending' | 'educational' | 'behind-
scenes' | 'entertainment';

    // Content types
    contentSplit: {
        feed: number; // % regular feed posts
        stories: number; // % story content
        reels: number; // % reel content
    }
}

```

```

    igtv: number; // % long-form video
};

// Engagement approach
captionLength: 'short' | 'medium' | 'storytelling' |
'novel-length';
hashtagStrategy: 'niche' | 'broad' | 'trending' |
'community';
commentStyle: 'emoji-heavy' | 'conversational' |
'supportive' | 'minimal';
}

// Similar detailed interfaces for TikTok, LinkedIn,
Facebook...

```

## Behavioral Patterns & Algorithms

### Authentic Behavior Simulation

```

interface BehavioralPatterns {
    // Posting behavior
    postingPatterns: PostingBehavior;

    // Engagement behavior
    engagementPatterns: EngagementBehavior;

    // Response patterns
    responsePatterns: ResponseBehavior;

    // Seasonal/temporal patterns
    temporalPatterns: TemporalBehavior;
}

interface PostingBehavior {
    // Timing variations
    timingVariation: {
        earliestPost: string; // '06:00'
        latestPost: string; // '23:30'
        preferredHours: string[]; // ['09:00', '12:00',
'18:00']
        randomnessMinutes: number; // ±15 minutes variation
    }
}

```

```

};

// Content planning
planningStyle: 'spontaneous' | 'planned' | 'scheduled' |
'mixed';
contentPreparation: {
  draftsUsage: 'never' | 'sometimes' | 'always';
  editingTime: 'quick' | 'moderate' | 'perfectionist';
  secondThoughts: number; // % of posts that get deleted/
edited
};

// Posting consistency
consistency: {
  streakTendency: number; // Likelihood of posting
multiple days in a row
  breakTendency: number; // Likelihood of taking breaks
  averageBreakDays: number; // Typical break length
};
}

interface EngagementBehavior {
  // Response timing
  responseDelays: {
    immediate: number; // % responses within 5 minutes
    quick: number; // % responses within 1 hour
    delayed: number; // % responses within 24 hours
    late: number; // % responses after 24 hours
  };

  // Engagement types
  engagementTypes: {
    likes: EngagementFrequency;
    comments: EngagementFrequency;
    shares: EngagementFrequency;
    saves: EngagementFrequency;
    directMessages: EngagementFrequency;
  };

  // Comment behavior

```

```

    commentBehavior: {
      length: 'emoji' | 'short' | 'medium' | 'detailed';
      tone: 'supportive' | 'questioning' | 'funny' |
'informative';
      followUp: number; // % likelihood of follow-up comments
    };
  }
}

```

```

type EngagementFrequency = 'never' | 'rare' | 'occasional'
| 'frequent' | 'very-frequent';

```

## Content Theme System

### Thematic Content Organization

```

interface ContentThemes {
  // Primary themes (60-80% of content)
  primaryThemes: ThemeDefinition[];

  // Secondary themes (15-25% of content)
  secondaryThemes: ThemeDefinition[];

  // Occasional themes (5-15% of content)
  occasionalThemes: ThemeDefinition[];

  // Avoided themes
  avoidedThemes: string[];

  // Seasonal/event themes
  seasonalThemes: SeasonalTheme[];
}

interface ThemeDefinition {
  name: string; // 'Personal Growth', 'Local Food',
'Technology'
  description: string;
  keywords: string[]; // Associated hashtags and keywords
  contentTypes: string[]; // 'story', 'tips', 'review',
'opinion'
  platforms: string[]; // Which platforms this theme works
on

```

```

    frequency: ThemeFrequency;
    examples: string[]; // Example post ideas
}

interface ThemeFrequency {
    postsPerWeek: number;
    preferredDays: string[];
    preferredTimes: string[];
    seasonalVariation: {
        spring: number; // Multiplier for frequency
        summer: number;
        fall: number;
        winter: number;
    };
}

interface SeasonalTheme {
    name: string; // 'Christmas', 'Back to School', 'Summer Travel'
    startDate: string; // 'MM-DD'
    endDate: string; // 'MM-DD'
    contentIdeas: string[];
    hashtagSets: string[][];
    intensityLevel: 'light' | 'moderate' | 'heavy'; // How much to focus on this theme
}

```

## Performance Metrics & Analytics

### Persona Performance Tracking

```

interface PersonaMetrics {
    // Basic stats
    totalPosts: number;
    totalEngagement: number;
    averageEngagementRate: number;

    // Platform-specific metrics
    platformMetrics: {
        [platform: string]: PlatformMetrics;
    };
}

```

```

// Content performance
contentPerformance: ContentPerformanceMetrics;

// Growth metrics
growthMetrics: GrowthMetrics;

// Risk metrics
riskMetrics: RiskMetrics;

// Authenticity score
authenticityScore: number; // 0-100, how authentic the
persona appears
}

interface PlatformMetrics {
  followers: number;
  following: number;
  posts: number;
  avgLikes: number;
  avgComments: number;
  avgShares: number;
  engagementRate: number;
  reachMetrics: ReachMetrics;
}

interface ContentPerformanceMetrics {
  topPerformingThemes: ThemePerformance[];
  bestPostingTimes: TimePerformance[];
  contentTypePerformance: ContentTypePerformance[];
  hashtagEffectiveness: HashtagPerformance[];
}

interface GrowthMetrics {
  followerGrowthRate: number; // Monthly %
  engagementGrowthRate: number;
  reachGrowthRate: number;
  goalProgress: GoalProgress[];
}

```

```
interface RiskMetrics {
    suspiciousActivityFlags: number;
    platformWarnings: number;
    accountLimitations: number;
    contentRemovals: number;
    overallRiskScore: number; // 0-100, higher = more risky
}
```

## **Persona Validation & Quality Assurance**

### **Character Consistency Validation**

```
interface PersonaValidation {
    // Consistency checks
    characterConsistency: ConsistencyChecks;

    // Content alignment
    contentAlignment: ContentAlignmentChecks;

    // Platform authenticity
    platformAuthenticity: PlatformAuthenticityChecks;

    // Performance validation
    performanceValidation: PerformanceChecks;
}

interface ConsistencyChecks {
    demographicConsistency: boolean; // Age, location,
background align
    personalityConsistency: boolean; // Traits consistent
across content
    languageConsistency: boolean; // Speaking style remains
consistent
    knowledgeConsistency: boolean; // References align with
background
    timelineConsistency: boolean; // Life events make
chronological sense
}

interface ContentAlignmentChecks {
```

```

    themeAlignment: number; // % of content matching defined
themes
    toneAlignment: number; // % of content matching
personality tone
    qualityConsistency: number; // Content quality remains
consistent
    engagementAlignment: number; // Engagement style matches
definition
}

interface ValidationRules {
    // Age-appropriate content rules
    ageConsistency: {
        minAge: number;
        maxAge: number;
        restrictedTopics: string[]; // Topics inappropriate for
this age
        requiredKnowledge: string[]; // Knowledge expected at
this age
    };

    // Location-appropriate content rules
    locationConsistency: {
        localReferences: string[]; // Must occasionally
reference local places
        culturalAppropriate: string[]; // Must align with local
culture
        timezoneConsistent: boolean; // Posting times match
location
    };

    // Professional consistency
    professionalConsistency: {
        careerKnowledge: string[]; // Must demonstrate career-
related knowledge
        industryAwareness: string[]; // Should know industry
trends
        networkingBehavior: boolean; // Networking style
matches profession
    };

```



```
}
```

## Persona Creation Workflow

### Step-by-Step Persona Development

```
interface PersonaCreationWorkflow {
  steps: PersonaCreationStep[];
  validation: ValidationStep[];
  finalization: FinalizationStep[];
}

interface PersonaCreationStep {
  stepNumber: number;
  stepName: string;
  description: string;
  requiredFields: string[];
  optionalFields: string[];
  validationRules: ValidationRule[];
  dependencies: string[]; // Which previous steps must be
complete
}

// Example workflow steps
const personaCreationSteps: PersonaCreationStep[] = [
  {
    stepNumber: 1,
    stepName: 'Basic Identity',
    description: 'Define the fundamental character
identity',
    requiredFields: ['penName', 'age', 'gender',
'primaryLocation'],
    optionalFields: ['nickname', 'culturalBackground'],
    validationRules: [
      { field: 'age', rule: 'between', min: 18, max: 80 },
      { field: 'penName', rule: 'unique', scope: 'global' }
    ],
    dependencies: []
  },
  {
    stepNumber: 2,
```

```

        stepName: 'Background & History',
        description: 'Develop the character backstory and life
experience',
        requiredFields: ['occupation', 'educationLevel',
'familyStatus'],
        optionalFields: ['previousJobs', 'travelHistory',
'majorLifeEvents'],
        validationRules: [
            { field: 'occupation', rule: 'appropriate_for_age' },
            { field: 'educationLevel', rule:
'consistent_with_age_and_career' }
        ],
        dependencies: ['Basic Identity']
    },
    // ... additional steps
];

```

## Export & Import System

### Persona Data Portability

```

interface PersonaExport {
    version: string; // Schema version for compatibility
    exportDate: string;
    personas: ExportedPersona[];
    metadata: ExportMetadata;
}

interface ExportedPersona {
    // Full persona data
    personaData: PenNamePersona;

    // Associated content templates
    contentTemplates: ContentTemplate[];

    // Performance history
    performanceHistory: PerformanceSnapshot[];

    // Platform connections (credentials excluded for
security)
    platformConnections: PlatformConnectionSummary[];
}

```

```
}
```

```
interface PersonaImport {  
    validateSchema(data: any): ValidationResult;  
    importPersona(data: ExportedPersona):  
    Promise<ImportResult>;  
    mergeConflicts(existing: PenNamePersona, imported:  
    ExportedPersona): ConflictResolution;  
}
```

This comprehensive persona schema provides the foundation for creating consistent, authentic, and maintainable pen name characters across multiple social media platforms while ensuring each persona remains believable and engaging.