Project Kage - Development Overview

Introduction

Project Kage is an AI-powered habit formation and goal tracking application designed to help users break free from digital addiction and build sustainable positive habits using neuroscience principles (DOSE: Dopamine, Oxytocin, Serotonin, Endorphins). This document provides a comprehensive roadmap for developers to understand what components to build first and how they interconnect.

Development Priorities & Phasing

Phase 1: Core Infrastructure (Months 1-2)

- Backend architecture setup
- Database design and implementation
- Authentication system
- Basic API framework
- DevOps pipeline configuration

Phase 2: MVP Core Features (Months 2-4)

1. User account and profile management
2. AI assistant core functionality
3. Goal setting and tracking framework
4. Basic habit tracking system
5. Simple journaling capability
6. Minimal scheduling functionality

Phase 3: Enhanced Features (Months 4-6)

1. Advanced AI assistant capabilities
2. Expanded goal breakdown methodology
3. Enhanced habit analytics and visualization
4. Journal insights and analysis
5. Intelligent scheduling suggestions
6. Basic community features

Phase 4: Polishing & Expansion (Months 6-7)

1. UI/UX refinement
2. Performance optimization

3. Cross-platform testing
4. Content library development
5. Premium feature implementation
6. Final pre-launch QA

Technical Architecture

1. Frontend Architecture

Mobile Applications

- **Framework**: React Native with TypeScript
- **State Management**: Redux with Redux Toolkit
- **UI Component Library**: Custom design system with base in React Native Paper
- **Navigation**: React Navigation
- **Styling**: Styled Components + design tokens

Web Application (Future)

- **Framework**: React.js with Next.js
- **State Management**: Redux or Context API
- **Styling**: Tailwind CSS

2. Backend Architecture

API Layer

- **Framework**: Node.js with Express
- **API Design**: GraphQL with Apollo Server + some REST endpoints
- **Authentication**: JWT with refresh token pattern
- **Validation**: Joi or Zod

Services

- **Architecture**: Microservices-lite (modular monolith initially, with clear boundaries for future separation)
- **Core Services**:
  - User Service
  - Goal Service
  - Habit Service
  - Journal Service

- Schedule Service
- Community Service
- AI Assistant Service
- Analytics Service

Database Layer

- **Primary Database**: MongoDB (document store)
  - Collections for users, goals, habits, journal entries, etc.
- **Secondary Database**: PostgreSQL (for transactional data and analytics)
- **Caching Layer**: Redis for performance and real-time features

AI Infrastructure

- **NLP Framework**: Transformers-based system with custom fine-tuning
- **Conversational Engine**: Custom-built on top of a large language model
- **Machine Learning Pipeline**: TensorFlow for model training and inference
- **User Behavior Analysis**: Custom analytics engine for personalized insights

3. DevOps Infrastructure

- **Containerization**: Docker
- **Orchestration**: Kubernetes for production (optional for initial phases)
- **CI/CD**: GitHub Actions
- **Cloud Provider**: AWS or Google Cloud
- **Monitoring**: Datadog or New Relic + Sentry for error tracking

Key Components & Dependencies

1. AI Assistant Development (Highest Priority)

The AI assistant is the central differentiating feature of Project Kage. It needs to:

Technical Requirements

- Fine-tune a base large language model with domain-specific data
- Develop custom training datasets for:
  - Goal-setting conversations
  - Habit formation guidance using DOSE principles
  - Journaling analysis
  - Schedule optimization

- Implement conversational memory and context management
- Create specialized modules for different assistant capabilities
- Develop an inference API optimized for mobile

Dependencies

- User profile data
- Goal and habit tracking data
- Journal content for analysis
- Schedule information

2. Goal Tracking System

Technical Requirements

- Implement goal creation and storage with hierarchical structure
- Develop the "5 Whys" methodology as guided conversation flow
- Create goal templates and recommendation system
- Build progress visualization components
- Implement milestone tracking functionality
- Create goal-habit connection mapping

Dependencies

- AI Assistant for guided goal setup
- Habit system for connecting goals to daily actions
- Analytics service for progress tracking

3. Habit Formation System

Technical Requirements

- Build habit tracking dashboard and data model
- Implement streak visualization and analytics
- Create difficulty adjustment algorithms
- Develop time-tracking functionality for duration-based habits
- Implement location-based triggering system
- Build comprehensive habit statistics

Dependencies

- Goal system for parent goal connections

- AI Assistant for habit recommendations
- Scheduling system for time-based habits

4. Scheduling System

Technical Requirements

- Implement time-blocking calendar with visual interface
- Develop schedule optimization algorithms
- Create daily routine view with status tracking
- Build intelligent habit scheduling recommendation engine
- Implement calendar service integrations (Google, Apple, Outlook)
- Create notification and reminder system

Dependencies

- Habit system for scheduling habits
- AI Assistant for optimization recommendations
- User preferences for timing suggestions

5. Journaling System

Technical Requirements

- Implement journal entry creation and storage
- Develop guided journaling prompts database
- Create mood and energy tracking functionality
- Build journal analysis algorithms for pattern recognition
- Implement goal and habit reflection journaling templates

Dependencies

- AI Assistant for analysis and insights
- Goal and habit systems for reflection connections
- Mood tracking analytics

6. Community Features

Technical Requirements

- Implement privacy-focused sharing system
- Build accountability partnership matching and management

- Create challenge participation framework
- Develop knowledge sharing and resource library
- Implement success story sharing with templates

Dependencies

- User profile system
- Goal and habit systems for sharing context
- Content moderation tools

Data Models (Core Schemas)

User

```javascript
{
  id: String,
  email: String,
  password: String (hashed),
  displayName: String,
  profilePicture: String,
  preferences: {
    aiAssistantStyle: String,
    notificationSettings: Object,
    themePreference: String,
    privacySettings: Object
  },
  subscriptionStatus: String,
  createdAt: Date,
  lastActive: Date
}
```

Goal

javascript

```
{
  id: String,
  userId: String,
  title: String,
  description: String,
  category: String,
  importance: Number,
  whys: [String], // 5 Whys responses
  startDate: Date,
  targetDate: Date,
  completionCriteria: String,
  progress: Number,
  status: String,
  milestones: [{
    id: String,
    title: String,
    description: String,
    dueDate: Date,
    completedDate: Date,
    status: String
  }],
  parentGoalId: String,
  subGoals: [String], // IDs of child goals
  relatedHabits: [String], // IDs of connected habits
  metrics: [{
    name: String,
    type: String,
    target: Number,
    current: Number
  }],
  reflections: [String], // IDs of journal entries
  visualizationType: String,
  createdAt: Date,
  updatedAt: Date
}
```

Habit

javascript

```
{
  id: String,
  userId: String,
  title: String,
  description: String,
  category: String,
  frequency: {
    type: String, // daily, weekly, etc.
    days: [Number], // days of week
    times: [Date] // time of day
  },
  cue: {
    type: String,
    details: Object
  },
  location: {
    enabled: Boolean,
    places: [Object]
  },
  duration: {
    required: Boolean,
    targetMinutes: Number
  },
  difficulty: Number,
  relatedGoalId: String,
  streak: {
    current: Number,
    longest: Number,
    history: [Object]
  },
  completionHistory: [{
    date: Date,
    status: String,
    notes: String
  }],
  doseCategory: String, // which neurochemical this primarily targets
  createdAt: Date,
  updatedAt: Date
}
```

Journal

```javascript
{
  id: String,
  userId: String,
  date: Date,
  content: String,
  template: String,
  mood: {
    primary: String,
    intensity: Number
  },
  energy: Number,
  tags: [String],
  location: String,
  weather: Object,
  relatedGoals: [String],
  relatedHabits: [String],
  aiInsights: [{
    type: String,
    content: String,
    generatedAt: Date
  }],
  isPrivate: Boolean,
  createdAt: Date,
  updatedAt: Date
}
```

Schedule

```javascript
{
  id: String,
  userId: String,
  date: Date,
  blocks: [{
    id: String,
    title: String,
    description: String,
    startTime: Date,
    endTime: Date,
    category: String,
    priority: Number,
    relatedHabitId: String,
    relatedGoalId: String,
    status: String,
    recurrence: Object
  }],
  energyPattern: {
    morning: Number,
    afternoon: Number,
    evening: Number
  },
  optimizationSuggestions: [Object],
  createdAt: Date,
  updatedAt: Date
}
```

AI Assistant Training Approach

The AI assistant is central to Project Kage. Here's a phased approach to its development:

Phase 1: Base Model Selection & Initial Training

- Start with an existing high-quality LLM (e.g., GPT, Claude, or open-source alternative)
- Develop initial prompt engineering for core conversational flows
- Create basic training datasets for habit formation and goal setting dialogues

Phase 2: Domain-Specific Fine-Tuning

- Curate specialized training data for DOSE principles
- Develop training datasets for habit coaching conversations
- Create goal breakdown methodology training examples
- Build journal analysis training examples

Phase 3: Persona Development & Interaction Styles

- Refine assistant personality to be supportive yet firm
- Develop different interaction styles based on user preferences
- Train response variation models for maintaining engagement
- Implement appropriate tone for setbacks and challenges

Phase 4: Integration & Contextualization

- Develop context-awareness capabilities using user data
- Build personalization models that learn from user interactions
- Create proactive suggestion algorithms
- Implement memory management for long-term relationships

Phase 5: Continuous Improvement

- Implement feedback loops for ongoing training
- Develop evaluation metrics for assistant effectiveness
- Create AB testing framework for new capabilities
- Build automated monitoring for interaction quality

First Components to Build

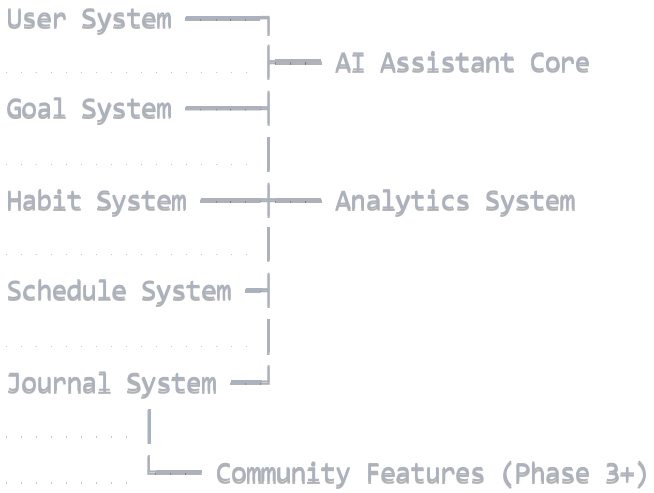
Based on dependency analysis and core value proposition, here are the first components to develop:

1. **User Account System**
   - Authentication and authorization
   - Profile management
   - Preference settings
   - Subscription handling
2. **AI Assistant Core**
   - Base conversational infrastructure
   - Initial NLP capabilities
   - DOSE principles knowledge base
   - Basic habit and goal guidance flows
3. **Goal Tracking Foundation**
   - Goal creation and storage
   - Basic goal visualization
   - "5 Whys" methodology implementation

- Goal templates library

4. **Habit Tracking Basics**
   - Habit creation and management
   - Streak visualization
   - Basic completion tracking
   - Habit-goal connections
5. **Simple Scheduling**
   - Time blocking interface
   - Schedule visualization
   - Habit time assignment
   - Basic notifications
6. **Minimal Journaling**
   - Journal entry creation
   - Simple prompted templates
   - Basic mood tracking
   - Goal/habit reflection templates

Integration Points & Dependencies Map



Technical Challenges & Considerations

1. **AI Performance on Mobile**
   - Optimize model size for mobile devices
   - Implement efficient inference
   - Consider on-device vs. cloud processing balance
2. **Data Privacy & Security**

- Ensure GDPR/CCPA compliance
- Implement end-to-end encryption for sensitive data
- Create clear data retention policies

3. **Offline Functionality**
   - Design for intermittent connectivity
   - Implement local storage with sync
   - Prioritize critical features for offline use
4. **Scalability**
   - Design database schema for growth
   - Implement sharding strategy
   - Plan for user content growth
5. **AI Training Evolution**
   - Create feedback loops for model improvement
   - Implement evaluation metrics
   - Balance personalization with privacy

Next Steps for Development Team

1. Set up development environment and CI/CD pipeline
2. Implement core user authentication system
3. Design and implement database schemas
4. Create API framework for core services
5. Begin AI assistant development with focus on core DOSE principles
6. Implement basic goal and habit tracking functionality
7. Develop initial mobile app UI with React Native
8. Focus on creating seamless AI assistant chat interface

This overview provides a structured roadmap for development. As the project progresses, more detailed specifications will be created for each component.