**Candidate Workflow Configuration Implementation Plan**

**Overview**

This implementation plan outlines the phased approach to developing the Candidate Workflow Configuration module for the GlobalRx platform. The plan organizes development into logical phases to ensure systematic progress while allowing for testing and validation at each stage.

**Phase 1: Database Schema and Core Structure**

**Database Schema Design (2 weeks)**

1. **Core Tables:**
   * Workflow: Store workflow definitions and settings
   * WorkflowPackage: Junction table linking workflows to packages
   * WorkflowSection: Define application sections and their order
   * ComplianceDocument: Store compliance forms and their metadata
   * CommunicationTemplate: Store notification templates
   * WorkflowTranslation: Store multilingual content for workflows
2. **Schema Relationships:**
   * Establish relationships between Workflow and existing tables (Package, Service)
   * Create proper indexes for performance optimization
   * Define constraints to ensure data integrity
3. **Migration Scripts:**
   * Prepare database migration scripts
   * Create seed data for testing

**Core API Development (2 weeks)**

1. **CRUD Operations:**
   * Implement API endpoints for managing workflows
   * Create endpoints for section management
   * Develop endpoints for compliance document management
2. **Validation Layer:**
   * Implement request validation using Zod schemas
   * Create response standardization middleware
3. **Authentication & Authorization:**
   * Implement role-based access control for workflow management
   * Secure APIs with proper permission checks

**Phase 2: Admin Interface Development**

**Workflow Management UI (2 weeks)**

1. **Workflow List View:**
   * Create table view of existing workflows
   * Implement filtering and search capabilities
   * Add status indicators and basic metrics
2. **Workflow Creation/Edit Form:**
   * Develop main workflow configuration form
   * Implement package selection interface
   * Create workflow name and description fields
3. **Section Management:**
   * Develop UI for configuring section order
   * Create section visibility controls
   * Implement section dependency configuration

**Compliance Document Management (1 week)**

1. **Document Upload Interface:**
   * Create document upload component
   * Implement document type selection
   * Add metadata input fields
2. **Document Association:**
   * Develop UI for linking documents to services
   * Create location-based document assignment
   * Implement document presentation configuration

**Communication Configuration (1 week)**

1. **Reminder Settings:**
   * Create interface for configuring reminder schedules
   * Implement channel selection (email/SMS/both)
   * Develop maximum reminder limit controls
2. **Template Management:**
   * Create template editor for notifications
   * Implement variable insertion system
   * Develop preview functionality

**Application Lifecycle Settings (1 week)**

1. **Expiration Configuration:**
   * Create controls for setting application timeframes
   * Implement auto-close rule configuration
   * Develop extension policy settings

**Multilingual Configuration (2 weeks)**

1. **Language Settings:**
   * Create language selection interface
   * Implement default language configuration
   * Develop location-based language suggestion settings
2. **Translation Management:**
   * Create translation management interface
   * Implement import/export functionality
   * Develop translation status tracking

**Phase 3: Dynamic Form Engine**

**Form Generation Framework (3 weeks)**

1. **Dynamic Field Loading:**
   * Develop system to load fields based on service requirements
   * Implement location-based field adjustments
   * Create mechanism to pull from DataRX definitions
2. **Conditional Logic Engine:**
   * Create rules engine for field dependencies
   * Implement show/hide logic based on previous answers
   * Develop validation trigger system
3. **Form State Management:**
   * Implement form state persistence
   * Create progress tracking system
   * Develop section completion validation

**Integration Layer (2 weeks)**

1. **Webhook Configuration:**
   * Create webhook management interface
   * Implement endpoint testing functionality
   * Develop event trigger configuration
2. **API Integration:**
   * Create third-party API configuration
   * Implement authentication setup for external systems
   * Develop data mapping interface

**Phase 4: Testing and Optimization**

**Testing Strategy (2 weeks)**

1. **Unit Testing:**
   * Create tests for core API functionality
   * Implement tests for workflow logic
   * Develop validation testing
2. **Integration Testing:**
   * Create tests for API integrations
   * Implement end-to-end workflow tests
   * Develop cross-module testing
3. **User Acceptance Testing:**
   * Prepare test scenarios
   * Develop test data generation scripts
   * Create UAT documentation

**Performance Optimization (2 weeks)**

1. **API Optimization:**
   * Implement query optimization
   * Add caching for frequently accessed data
   * Develop batch operations for efficiency
2. **UI Performance:**
   * Optimize component rendering
   * Implement lazy loading
   * Add pagination for large data sets
3. **Database Tuning:**
   * Analyze and optimize queries
   * Add appropriate indexes
   * Implement query monitoring

**Phase 5: Documentation and Deployment**

**Documentation (1 week)**

1. **Technical Documentation:**
   * Create API documentation
   * Develop database schema documentation
   * Document integration points
2. **User Guide:**
   * Create administrator guide for workflow configuration
   * Develop quick reference materials
   * Create video tutorials for common tasks

**Deployment Planning (1 week)**

1. **Deployment Strategy:**
   * Create deployment checklist
   * Implement database migration strategy
   * Develop rollback procedures
2. **Feature Flags:**
   * Implement feature flag system
   * Configure gradual rollout
   * Create monitoring alerts

**Timeline and Resource Allocation**

**Total Estimated Timeline: 22 weeks**

1. **Phase 1 (Database Schema and Core Structure)**: 4 weeks
   * Database Developer (1)
   * API Developer (1)
2. **Phase 2 (Admin Interface Development)**: 7 weeks
   * Frontend Developer (2)
   * UI/UX Designer (1)
   * Backend Developer (1)
3. **Phase 3 (Dynamic Form Engine)**: 5 weeks
   * Frontend Developer (1)
   * Backend Developer (1)
   * Integration Specialist (1)
4. **Phase 4 (Testing and Optimization)**: 4 weeks
   * QA Engineer (2)
   * Performance Engineer (1)
   * Full-stack Developer (1)
5. **Phase 5 (Documentation and Deployment)**: 2 weeks
   * Technical Writer (1)
   * DevOps Engineer (1)
   * Project Manager (1)

**Key Dependencies**

1. **Database Schema**: Must be finalized before API development can be completed
2. **API Layer**: Required before admin interface development
3. **Package and Service Integration**: Dependent on existing modules
4. **DataRX Integration**: Requires access to field definitions and validation rules

**Risk Assessment and Mitigation**

**Key Risks:**

1. **Integration Complexity**:
   * **Risk**: Integration with DataRX and Package systems may be more complex than anticipated
   * **Mitigation**: Begin with API mocks, conduct early integration testing
2. **Performance Issues**:
   * **Risk**: Dynamic form generation may cause performance bottlenecks
   * **Mitigation**: Implement performance testing early, use lazy loading and pagination
3. **Scope Creep**:
   * **Risk**: Requirements may expand during development
   * **Mitigation**: Implement strict change control, prioritize core functionality
4. **Browser Compatibility**:
   * **Risk**: Complex UI may have compatibility issues
   * **Mitigation**: Implement cross-browser testing early, use standardized components

**Success Criteria**

1. Administrators can create and manage workflows with associated packages
2. Dynamic form generation correctly displays fields based on service requirements
3. Communication system properly sends notifications via configured channels
4. Multilingual support functions correctly for all configured languages
5. Performance meets established benchmarks (page load < 2s, form submission < 1s)
6. All test cases pass with 90%+ coverage

**Next Steps**

1. **Immediate Actions**:
   * Finalize database schema design
   * Create detailed technical specifications
   * Establish development environment
2. **Long-term Planning**:
   * Develop integration strategy with candidate portal
   * Plan for scalability as workflow volume increases
   * Prepare training materials for administrators