

# Software Requirements Specification (SRS) - MedCareer Platform

---

## 1. EXECUTIVE SUMMARY

**Project Name:** MedCareer - Medical Professional Job Recruitment Platform

**Version:** 2.0

**Date:** January 2025

**Purpose:** A specialized job recruitment platform designed specifically for healthcare professionals, emphasizing speed, efficiency, and user-friendly interfaces to minimize time spent on job searching and application processes.

## 2. PROJECT OVERVIEW

**Vision:** To create the most efficient and user-friendly job recruitment platform for healthcare professionals, reducing the time from job discovery to application submission to under 2 minutes.

**Mission:** Streamline the job search and application process for busy healthcare professionals who have limited time between shifts and patient care responsibilities.

## 3. STAKEHOLDER ANALYSIS

**Primary Users:**

- Healthcare Professionals (Job Seekers)
- Healthcare Employers/Recruiters
- Platform Administrators

**Key Pain Points Addressed:**

- Time constraints of healthcare professionals
- Complex application processes
- Lack of medical specialization filtering
- Poor mobile experience during breaks/commutes

## 4. FUNCTIONAL REQUIREMENTS

### 4.1 Core Features (Current Implementation)

#### User Authentication System

- Dual authentication (Employers & Job Seekers)
- JWT-based secure authentication
- Protected routes and role-based access

## ✅ Job Management

- Job posting with medical specializations
- Advanced filtering by specialization, location, job type
- Real-time job search functionality

## ✅ Profile Management

- Comprehensive job seeker profiles
- CV/Resume upload functionality
- Profile picture management

## ✅ Application System

- One-click job applications
- Application status tracking
- Compatibility scoring system

## 4.2 Enhanced Features (To Be Implemented)

### 🚀 Quick Apply System

- One-click application with pre-filled profile data
- Bulk application functionality for similar positions
- Smart application suggestions based on profile

### 📱 Mobile-First Design

- Progressive Web App (PWA) capabilities
- Offline job browsing
- Push notifications for new relevant jobs

### 🔍 Intelligent Job Matching

- AI-powered job recommendations
- Compatibility scoring enhancement
- Smart alerts for high-match positions

### ⚡ Speed Optimizations

- Instant search with debouncing
- Lazy loading for job listings
- Cached frequently accessed data

## 5. NON-FUNCTIONAL REQUIREMENTS

## 5.1 Performance Requirements

- Page load time: < 2 seconds
- Search response time: < 500ms
- Application submission: < 1 second
- 99.9% uptime availability

## 5.2 Usability Requirements

- Maximum 3 clicks to apply for a job
- Mobile-responsive design (< 768px screens)
- Accessibility compliance (WCAG 2.1 AA)
- Multi-language support (English, French, Arabic)

## 5.3 Security Requirements

- HTTPS encryption for all communications
- JWT token expiration and refresh
- File upload security (CV/Resume validation)
- GDPR compliance for data protection

# 6. SYSTEM ARCHITECTURE

## 6.1 Current Technology Stack

### Frontend:

- React 18 with Vite
- Material-UI & Tailwind CSS
- React Router v6
- React Query for state management
- Axios for API calls

### Backend:

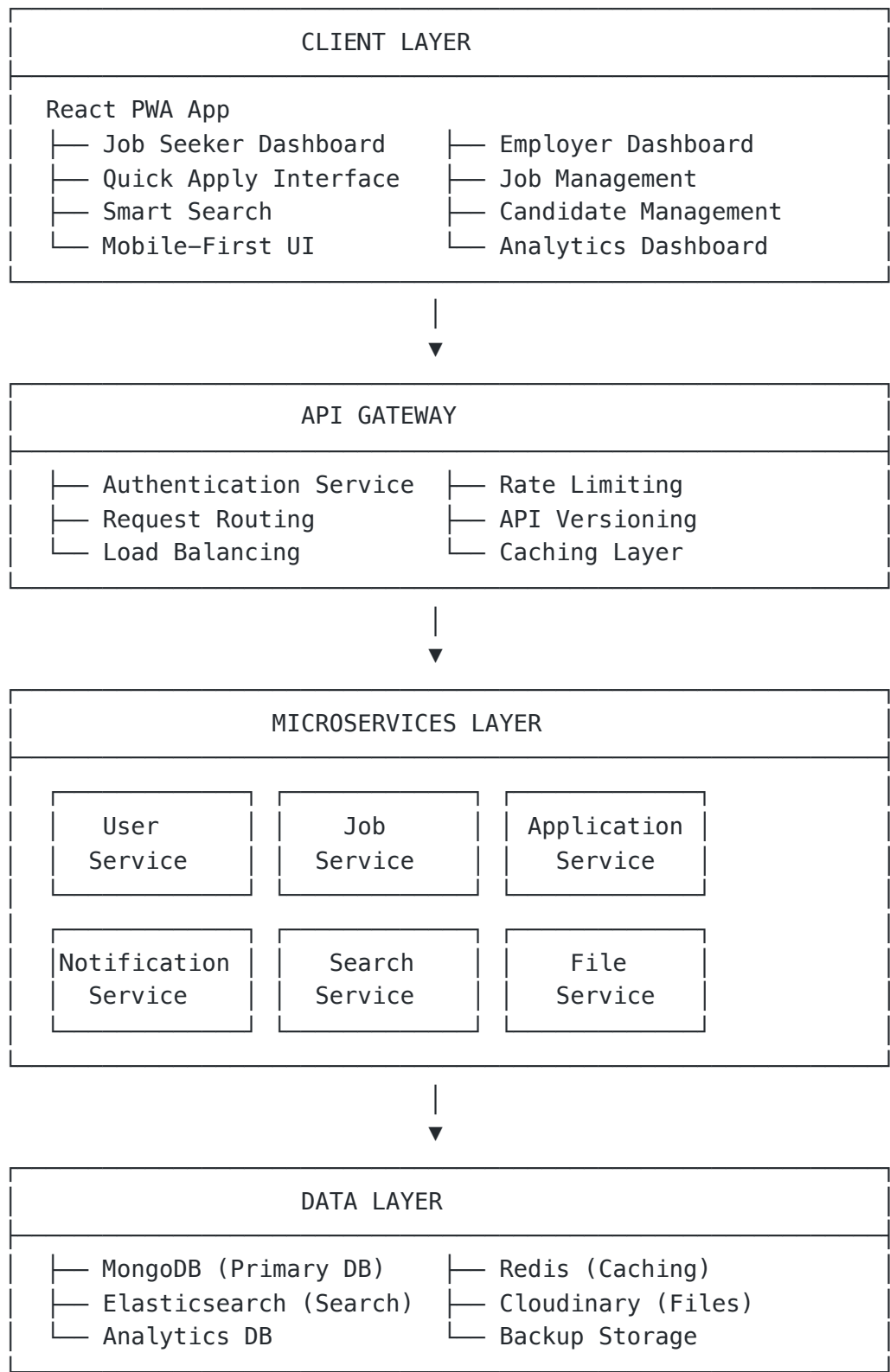
- Node.js with Express.js
- MongoDB with Mongoose ODM
- JWT for authentication
- Cloudinary for file storage
- Multer for file uploads

### Infrastructure:

- RESTful API architecture

- MVC pattern implementation
- Middleware-based request processing

6.2 Proposed Architecture Enhancements



7. IMPLEMENTATION PLAN

Phase 1: Foundation & Optimization (Weeks 1-4)

## **Week 1-2: Performance Optimization**

- Implement React Query for better state management
- Add lazy loading for job listings
- Optimize bundle size with code splitting
- Implement service worker for PWA

## **Week 3-4: UI/UX Enhancement**

- Redesign job search interface for mobile-first
- Implement quick apply functionality
- Add advanced filtering with saved searches
- Improve accessibility features

## **Phase 2: Smart Features (Weeks 5-8)**

### **Week 5-6: Intelligent Matching**

- Implement job recommendation algorithm
- Add compatibility scoring enhancements
- Create smart notification system
- Build job alert functionality

### **Week 7-8: Advanced Search**

- Integrate Elasticsearch for faster search
- Add autocomplete and search suggestions
- Implement search analytics
- Create saved search functionality

## **Phase 3: Mobile & Performance (Weeks 9-12)**

### **Week 9-10: Mobile Optimization**

- Complete PWA implementation
- Add offline functionality
- Implement push notifications
- Optimize for various screen sizes

### **Week 11-12: Analytics & Admin**

- Build comprehensive admin dashboard
- Add user behavior analytics
- Implement A/B testing framework

- Create performance monitoring

## 8. DATABASE SCHEMA ENHANCEMENTS

```
// Enhanced JobSeeker Schema
const JobSeekerSchema = {
  // Existing fields...
  preferences: {
    jobTypes: [String],
    specializations: [String],
    locations: [String],
    salaryRange: { min: Number, max: Number },
    workSchedule: [String] // full-time, part-time, night-shift, etc.
  },
  quickApplyEnabled: { type: Boolean, default: true },
  searchHistory: [{ query: String, timestamp: Date }],
  savedJobs: [{ type: ObjectId, ref: 'Job' }],
  applicationStats: {
    totalApplications: Number,
    responseRate: Number,
    averageResponseTime: Number
  }
}

// Enhanced Job Schema
const JobSchema = {
  // Existing fields...
  urgency: { type: String, enum: ['low', 'medium', 'high', 'urgent'] },
  benefits: [String],
  requirements: {
    experience: String,
    education: String,
    certifications: [String],
    skills: [String]
  },
  salaryRange: { min: Number, max: Number },
  workSchedule: String,
  applicationDeadline: Date,
  quickApplyEnabled: { type: Boolean, default: true }
}
```

## 9. API ENDPOINTS ENHANCEMENT

```
// Quick Apply Endpoints
POST /api/v1/applications/quick-apply/:jobId
POST /api/v1/applications/bulk-apply
GET /api/v1/jobs/recommendations/:jobSeekerId

// Search & Filter Endpoints
```

```
GET /api/v1/jobs/search?q=:query&filters=:filters
GET /api/v1/jobs/autocomplete?q=:query
POST /api/v1/jobs/save-search
GET /api/v1/jobs/saved-searches
```

```
// Notification Endpoints
POST /api/v1/notifications/subscribe
GET /api/v1/notifications/preferences
PUT /api/v1/notifications/preferences
```

## 10. SECURITY CONSIDERATIONS

- **Data Encryption:** All sensitive data encrypted at rest and in transit
- **Input Validation:** Comprehensive validation for all user inputs
- **File Security:** Virus scanning for uploaded CVs and documents
- **Rate Limiting:** API rate limiting to prevent abuse
- **GDPR Compliance:** Data retention policies and user data export/deletion

## 11. TESTING STRATEGY

- **Unit Testing:** Jest for component and utility testing
- **Integration Testing:** API endpoint testing with Supertest
- **E2E Testing:** Cypress for user journey testing
- **Performance Testing:** Load testing with Artillery
- **Mobile Testing:** Cross-device testing with BrowserStack

## 12. DEPLOYMENT & MONITORING

- **CI/CD Pipeline:** GitHub Actions for automated deployment
- **Containerization:** Docker for consistent environments
- **Monitoring:** Application performance monitoring with New Relic
- **Error Tracking:** Sentry for error monitoring and alerting
- **Analytics:** Google Analytics and custom event tracking

## 13. SUCCESS METRICS

- **User Engagement:** Average session duration > 5 minutes
- **Application Rate:** > 80% of job views result in applications
- **Time to Apply:** Average application time < 2 minutes
- **User Retention:** 70% monthly active user retention
- **Mobile Usage:** > 60% of traffic from mobile devices