SCOPE OF WORK (SOW)

Enout Event Management Application

Project: Enout Event Management App

Repository: https://github.com/Tanmay-enout/enout-event-management-APP-

Document Version: 1.0

Date: October 2024

---

1. COMPLETED DEVELOPMENT WORK

1.1 Admin Dashboard (Next.js/React)

Status: ✅ **FULLY IMPLEMENTED**

Core Features:

- **Event Management System**
    - Create, read, update, delete events
    - Event status calculation (pending/in_progress/complete)
    - Event search and filtering
    - Event selection and navigation

    - Fallback events for development
- **Guest Management System**
    - Guest list with pagination (100 items per page)
    - Guest creation, editing, and deletion
    - Bulk actions (send invites, resend invites, export CSV)
    - Guest filtering and search
    - Import functionality for CSV files
    - Guest status tracking (invited, registered, accepted)
- **Message Broadcasting System**
    - 3-tab layout: Sent, Drafts, Send Message
    - Message composer with rich text editing
    - Draft creation and editing
    - Message scheduling functionality
    - Audience targeting (all, invited, accepted)
- **Schedule Management**
    - Calendar view with week/month navigation
    - Schedule item creation, editing, deletion
    - Time slot management
    - Event timeline display
    - Optimized performance with memoization
- **Room Management**
    - Room assignment interface
    - Guest-to-room mapping
    - Room status tracking
    - Export functionality

UI/UX Components:

- Responsive dashboard layout
- Sidebar navigation with event list
- Top tabs for event sections
- Modal dialogs for forms
- Toast notifications
- Loading states and error handling
- Mobile-responsive design

State Management:

- Zustand for global state
- React Query for server state
- Local storage for temporary data
- Optimistic updates

1.2 Backend API (NestJS)

Status: ✅ FULLY IMPLEMENTED

Core Modules:

- **Events Module**
  - CRUD operations for events
  - Event filtering and pagination
  - Event status management
- **Authentication Module**
  - OTP-based authentication
  - JWT token management
  - Admin and attendee auth flows
  - Passwordless login system

- **Invites/Guests Module**
    - Guest invitation system
    - Bulk invite operations
    - Guest status tracking
    - CSV import/export
- **Schedule Module**
    - Itinerary item management
    - Schedule CRUD operations
    - Time-based filtering
- **Messages Module**
    - Message broadcasting
    - Draft management
    - Message scheduling
    - Audience targeting
- **Attendees Module**
    - Attendee profile management
    - Check-in functionality
    - Status tracking

Infrastructure:

- **Database Integration**
    - Prisma ORM with PostgreSQL
    - Database migrations
    - Seed data for development
    - Connection pooling
- **Security Features**

  - CORS configuration
  - Rate limiting (configurable)
  - Input validation with DTOs
  - Error handling middleware
  - Helmet security headers
- **API Documentation**
  - Swagger/OpenAPI integration
  - Comprehensive endpoint documentation
  - Request/response schemas

1.3 Mobile Application (React Native/Expo)

Status: 🔄 PARTIALLY IMPLEMENTED

Implemented Features:

- **Authentication Flow**
  - Email OTP verification
  - Phone number verification
  - Token-based authentication
  - Mock API integration
- **Task Management**
  - Task completion tracking
  - ID card upload
  - Registration form
  - Phone verification
- **Navigation Structure**
  - Expo Router implementation
  - Public and authenticated routes

    - Deep linking support

UI Components:

- Responsive mobile layouts
- Form components
- Task completion indicators
- Development data banner

1.4 Database Schema

Status: ✅ FULLY IMPLEMENTED

Core Models:

- **User Management**
    - Users with role-based access
    - Admin and HR roles
    - Authentication tracking
- **Event Management**
    - Events with full metadata
    - Timezone support
    - Status tracking
    - Location management
- **Guest Management**
    - Invites with status tracking
    - Attendee profiles
    - Bulk operations support
- **Schedule Management**
    - Itinerary items
    - Time-based scheduling

    - Location tracking
- **Room Management**
    - Room assignments
    - Guest-to-room mapping
    - Room status tracking
- **Message System**
    - Message broadcasting
    - Draft management
    - Audience targeting

---

2. PENDING/INCOMPLETE WORK

2.1 Mobile Application

Status: 🔄 REQUIRES COMPLETION

Missing Features:

- **Real API Integration**
    - Currently using mock APIs only
    - Network calls not implemented
    - Storage stubs need real implementation
- **Core Event Features**
    - Event browsing and details
    - Schedule viewing
    - Message inbox
    - Profile management
    - Check-in functionality
- **Production Features**

    - Push notifications
    - Offline support
    - Real-time updates
    - Image upload functionality

2.2 Backend API Enhancements

Status: 🔄 PARTIALLY COMPLETE

Missing Implementations:

- **Room API Endpoints**
    - Room CRUD operations
    - Room assignment logic
    - Room status management
- **Message Broadcasting**
    - Email integration (currently disabled)
    - SMS notifications
    - Push notification service
- **File Upload System**
    - Image upload for events
    - Document upload for guests
    - File storage integration

2.3 Production Readiness

Status: ❌ NOT STARTED

Missing Infrastructure:

- **Environment Configuration**
    - Production environment setup

    - Environment-specific configurations
    - Secrets management
- **Deployment Pipeline**
    - CI/CD configuration
    - Docker containerization
    - Cloud deployment setup
- **Monitoring & Logging**
    - Application monitoring
    - Error tracking
    - Performance monitoring

---

3. KNOWN ISSUES & BUGS

3.1 Critical Issues

Priority: 🔴 HIGH

1. **Rate Limiting Disabled**
    - `RATE_LIMIT_ENABLED="false"` in development
    - Needs proper configuration for production
    - **Impact:** Security vulnerability
2. **Email Service Disabled**
    - `EMAIL_ENABLED="false"` in configuration
    - Message broadcasting won't work
    - **Impact:** Core functionality broken
3. **Mock Data Dependencies**
    - Mobile app relies entirely on mocks
    - Admin dashboard has fallback events

- **Impact:** No real data flow

3.2 Performance Issues

Priority: 🟡 **MEDIUM**

1. **API Rate Limiting**
    - 429 errors during development
    - Retry logic implemented but needs tuning
    - **Impact:** Poor user experience
2. **Database Queries**
    - Some queries lack proper indexing
    - N+1 query problems in some endpoints
    - **Impact:** Slow response times
3. **Frontend Performance**
    - Large bundle sizes
    - Unoptimized images
    - **Impact:** Slow loading times

3.3 UI/UX Issues

Priority: 🟡 **MEDIUM**

1. **Error Handling**
    - Inconsistent error messages
    - Some errors not user-friendly
    - **Impact:** Poor user experience
2. **Loading States**
    - Some components lack loading indicators
    - Skeleton screens not implemented
    - **Impact:** Perceived performance issues

3. **Mobile Responsiveness**
    - Some admin components not mobile-optimized
    - Touch interactions need improvement
    - **Impact:** Mobile usability issues

---

4. TECHNICAL DEBT & IMPROVEMENTS

4.1 Code Quality

Priority: 🟡 MEDIUM

1. **TypeScript Improvements**
    - Some `any` types need proper typing
    - Missing interface definitions
    - **Action:** Add comprehensive type definitions
2. **Error Handling**
    - Inconsistent error handling patterns
    - Missing error boundaries
    - **Action:** Implement global error handling
3. **Code Organization**
    - Some components are too large
    - Business logic mixed with UI
    - **Action:** Refactor into smaller, focused components

4.2 Architecture Improvements

Priority: 🟡 MEDIUM

1. **API Client**
    - Retry logic needs refinement
    - Timeout handling could be improved

- **Action:** Implement robust API client

2. **State Management**
   - Some state updates could be optimized
   - Cache invalidation needs improvement
   - **Action:** Optimize state management patterns
3. **Database Schema**
   - Some relationships could be optimized
   - Missing indexes on frequently queried fields
   - **Action:** Add database optimizations

4.3 Security Improvements

Priority: 🔴 HIGH

1. **Authentication**
   - JWT secret is hardcoded
   - Token expiration handling needs improvement
   - **Action:** Implement proper secret management
2. **Input Validation**
   - Some endpoints lack proper validation
   - SQL injection prevention needs review
   - **Action:** Strengthen input validation
3. **CORS Configuration**
   - Currently allows all origins in development
   - **Action:** Configure proper CORS for production

---

5. NEXT STEPS & RECOMMENDATIONS

5.1 Immediate Priorities

Priority: 🔴 CRITICAL

1. **Enable Production Services**
    - Configure email service with real provider
    - Enable rate limiting with proper limits
    - Set up production database
    - **Effort:** 2-3 days
2. **Fix Critical Bugs**
    - Resolve 429 rate limiting errors
    - Fix message broadcasting
    - Implement proper error handling
    - **Effort:** 3-4 days
3. **Complete Mobile API Integration**
    - Replace mock APIs with real endpoints
    - Implement proper storage
    - Add network error handling
    - **Effort:** 5-7 days

5.2 Short-term Goals

Priority: 🟡 HIGH

1. **Production Deployment**
    - Set up CI/CD pipeline
    - Configure production environment
    - Implement monitoring
    - **Effort:** 5-7 days
2. **Performance Optimization**
    - Optimize database queries

- Implement caching
- Bundle size optimization
- **Effort:** 3-5 days

3. **Testing Implementation**
   - Unit tests for critical functions
   - Integration tests for APIs
   - E2E tests for user flows
   - **Effort:** 7-10 days

5.3 Medium-term Goals

Priority: 🟢 MEDIUM

1. **Feature Enhancements**
   - Advanced reporting
   - Analytics dashboard
   - Bulk operations improvements
   - **Effort:** 10-14 days
2. **Mobile App Completion**
   - Real-time notifications
   - Offline support
   - Advanced features
   - **Effort:** 14-21 days
3. **Security Hardening**
   - Security audit
   - Penetration testing
   - Compliance review
   - **Effort:** 7-10 days

5.4 Long-term Goals

Priority: 🟢 **LOW**

1. **Scalability Improvements**
    - Microservices architecture
    - Load balancing
    - Database sharding
    - **Effort:** 21-30 days
2. **Advanced Features**
    - AI-powered recommendations
    - Advanced analytics
    - Third-party integrations
    - **Effort:** 30+ days

---

6. DEVELOPMENT ENVIRONMENT SETUP

6.1 Prerequisites

- Node.js 18+
- PostgreSQL 14+
- Redis 6+
- pnpm package manager

6.2 Quick Start Commands

```
# Install dependencies
pnpm install

# Set up database
cd apps/api
pnpm prisma:generate
pnpm prisma:push
pnpm prisma:seed

# Start development servers
pnpm --filter @enout/api dev      # API on port 3006
pnpm --filter @enout/admin dev    # Admin on port 3000
pnpm --filter enout-mobile start  # Mobile app
```

6.3 Environment Variables

- Copy `.env.example` files
- Configure database URLs
- Set up email service credentials
- Configure JWT secrets

---

7. CONTACT & SUPPORT

Repository: https://github.com/Tanmay-enout/enout-event-management-APP-

Documentation: See README.md for detailed setup instructions

Issues: Use GitHub Issues for bug reports and feature requests

---

Document prepared by: Development Team

Last updated: October 2024

Next review: Upon project handover