TDIU Report Service: Client Portal Implementation Plan

Client Portal Overview

The client portal is a critical component of the TDIU Report Service, providing attorneys with a secure, user-friendly interface for uploading documents, tracking cases, and accessing reports. The portal will be implemented using AWS Amplify for the frontend, Amazon Cognito for authentication, and API Gateway for secure backend communication with Lambda functions.

Core Portal Requirements

1. **Security & Compliance**
   - HIPAA-compliant user authentication
   - Secure document transmission
   - Audit logging of all user activities
   - Data encryption in transit and at rest
2. **User Experience**
   - Intuitive interface for attorneys
   - Mobile-responsive design
   - Clear status indicators for case progress
   - Efficient document upload process
3. **Functionality**
   - Secure document uploads
   - Case status tracking
   - Report access and download
   - User profile management
   - Communication tools

Implementation Architecture

The client portal implementation will use the following AWS services:

1. AWS Amplify

- **Purpose**: Frontend hosting and development framework
- **Features**:
  - Responsive web application hosting

- CI/CD pipeline integration
- Built-in security features
- Integration with other AWS services

2. Amazon Cognito

- **Purpose**: User authentication and management
- **Features**:
  - Multi-factor authentication
  - User pool management
  - Identity federation
  - JWT token generation

3. API Gateway

- **Purpose**: Secure backend API for Lambda integration
- **Features**:
  - REST API endpoints
  - Request validation
  - Access control
  - Usage plans and throttling

4. Integration with Existing Components

- **S3 Buckets**: For document storage
- **Lambda Functions**: For document processing
- **CloudTrail**: For compliance logging

Portal Feature Implementation

1. Authentication System

Components:

- Cognito User Pools for attorney accounts
- Authentication UI components
- MFA implementation
- Password policies

- Session management

Implementation Steps:

1. Create Cognito User Pool with appropriate settings
2. Configure password policies (complexity, expiration)
3. Set up multi-factor authentication
4. Implement authentication UI components
5. Configure JWT token handling

2. Document Upload System

Components:

- Upload interface with drag-and-drop
- S3 pre-signed URL generation
- Upload progress tracking
- File validation
- Success/error handling

Implementation Steps:

1. Create upload UI component with drag-and-drop
2. Integrate with TDIU-GenerateUploadUrl Lambda
3. Implement direct-to-S3 uploads using pre-signed URLs
4. Add progress tracking and error handling
5. Create success notifications

3. Case Management Dashboard

Components:

- Case listing with status indicators
- Case detail view
- Document list per case
- Timeline visualization
- Status updates

Implementation Steps:

1. Design dashboard layout with case cards
2. Create case detail component
3. Implement status visualization
4. Add document listing for each case
5. Create filtering and sorting options

4. Report Access System

Components:

- Report listing
- Secure download mechanism
- Preview capability
- Version tracking
- Feedback submission

Implementation Steps:

1. Create report listing component
2. Implement secure download mechanism using pre-signed URLs
3. Add PDF preview capability
4. Create feedback submission form
5. Implement version tracking for revised reports

5. User Profile Management

Components:

- Profile information management
- Notification preferences
- Security settings
- Billing information (if applicable)
- Activity history

Implementation Steps:

1. Create profile management interface
2. Implement notification preference settings

3. Add security settings (password change, MFA)
4. Create activity history view
5. Add billing information management

UI/UX Design

Design Principles

- Clean, professional interface
- Intuitive navigation
- Clear hierarchy of information
- Accessible design
- Mobile-first approach

Key Screens

1. **Login Screen**
   - Simple, focused authentication
   - Password reset functionality
   - MFA integration
   - Branding elements
2. **Dashboard**
   - Summary of active cases
   - Quick actions
   - Status notifications
   - Recent activity
3. **Upload Interface**
   - Drag-and-drop area
   - File selection button
   - Progress indicators
   - File type validation
4. **Case Detail View**
   - Case information summary
   - Document listing
   - Status timeline

- Communication history
- Report access

5. **Report Viewer**
   - PDF preview
   - Download options
   - Feedback submission
   - Version history

Security Implementation

Authentication Security

- Enforced password complexity
- Multi-factor authentication
- Session timeout controls
- JWT token validation
- Login attempt limiting

Data Transmission Security

- TLS encryption for all communications
- Signed API requests
- Temporary credentials for uploads
- API Gateway request validation
- Content validation

Access Control

- Role-based access control
- Resource-level permissions
- JWT token validation
- API authorization
- Least privilege principle

Implementation Timeline

Phase 1: Core Authentication (Week 1)

- Set up AWS Amplify project
- Configure Cognito User Pool
- Create basic authentication UI
- Implement login/logout flow
- Test authentication security

Phase 2: Document Upload (Week 2)

- Create upload interface components
- Integrate with S3 pre-signed URLs
- Implement progress tracking
- Add validation and error handling
- Test upload functionality

Phase 3: Case Management (Week 3)

- Develop case dashboard
- Create case detail views
- Implement status tracking
- Add document listing
- Test case management flow

Phase 4: Report Access (Week 3-4)

- Create report listing interface
- Implement secure download
- Add feedback mechanism
- Create preview capability
- Test report access security

Phase 5: Finalization (Week 4)

- Integrate all components
- Implement remaining UI elements
- Conduct security testing
- Perform user acceptance testing
- Deploy to production

CloudFormation Integration

The client portal components will be added to the CloudFormation template to ensure complete infrastructure documentation:

yaml

```yaml
# Client Portal Resources
CognitoUserPool:
  Type: AWS::Cognito::UserPool
  Properties:
    UserPoolName: TDIU-UserPool
    AdminCreateUserConfig:
      AllowAdminCreateUserOnly: true
    AutoVerifiedAttributes:
      - email
    MfaConfiguration: "ON"
    Policies:
      PasswordPolicy:
        MinimumLength: 12
        RequireLowercase: true
        RequireNumbers: true
        RequireSymbols: true
        RequireUppercase: true

CognitoUserPoolClient:
  Type: AWS::Cognito::UserPoolClient
  Properties:
    ClientName: TDIU-App-Client
    GenerateSecret: false
    UserPoolId: !Ref CognitoUserPool
    ExplicitAuthFlows:
      - ALLOW_USER_PASSWORD_AUTH
      - ALLOW_REFRESH_TOKEN_AUTH

ApiGateway:
  Type: AWS::ApiGateway::RestApi
  Properties:
    Name: TDIU-API
    Description: API for TDIU Report Service
    EndpointConfiguration:
      Types:
        - REGIONAL

AmplifyApp:
  Type: AWS::Amplify::App
  Properties:
    Name: TDIU-Client-Portal
    Repository: https://github.com/your-repo/tdiu-client-portal
    BuildSpec: |-
```

```yaml
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm install
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: build
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
```

HTML/CSS Implementation

The client portal will be developed using React with the following structure:

1. **Component Architecture**
   - Reusable UI components
   - Container components for business logic
   - Context providers for state management
   - HOCs for cross-cutting concerns
2. **Styling Approach**
   - CSS modules for component styling
   - Responsive design system
   - Consistent color palette
   - Accessibility-first approach
3. **Key Components**
   - Authentication components
   - Upload components
   - Dashboard components
   - Case management components
   - Report components

Next Steps

1. **Amplify Project Setup**
   - Initialize Amplify project
   - Configure authentication
   - Set up basic infrastructure
2. **Authentication Implementation**
   - Create login/signup forms
   - Implement authentication flow
   - Add MFA configuration
3. **Upload Components Development**
   - Create document upload interface
   - Implement S3 integration
   - Add progress tracking
4. **Dashboard Implementation**
   - Develop case dashboard
   - Create status visualization
   - Implement filtering and sorting

The implementation will proceed according to the timeline, with each phase building upon the previous to create a comprehensive, secure client portal for the TDIU Report Service.