**Canadian Immigration, Refugees & Citizenship Department**

***Passport Automation System***

**1.0 Purpose of the Document**

The **System Architecture Documentation** provides an overview of the system's structure, components, and interactions for the **Passport Automation System**. This document ensures that all stakeholders understand the **technical framework, data flow, and infrastructure requirements** for implementation and maintenance.

**2.0 System Overview**

The **Passport Automation System** is a web and mobile-based platform that enables Canadian citizens to apply for, track, and receive their passports digitally. The system integrates with **government databases, police verification, and Canada Post for delivery**.

**3.0 High-Level Architecture**

The system follows a **three-tier architecture**:

1. **Presentation Layer (Frontend)**
   * Developed using **React Native** for mobile apps and **React.js** for the web.
   * Provides UI for passport application, document upload, and application tracking.
2. **Application Layer (Backend)**
   * Developed using **Spring Boot (Java)** for handling business logic.
   * Provides RESTful APIs for frontend integration.
   * Implements OAuth 2.0 authentication for user verification.
3. **Data Layer (Database & Storage)**
   * Uses **MySQL** for structured data storage (User, Passport, Payments, Verification).
   * Uses **Amazon S3** for document storage.
   * Includes logging mechanisms for audit tracking.

**4.0 Database Schema**

**4.1 Key Tables**

| **Table Name** | **Description** |
| --- | --- |
| Users | Stores applicant details (name, email, DOB, etc.). |
| Applications | Tracks passport application status. |
| Documents | Stores references to uploaded verification documents. |
| Payments | Manages transaction details for passport fees. |
| Verification | Logs police verification results. |

**5.0 Security Measures**

1. **Authentication & Authorization**
   * OAuth 2.0 with JWT tokens for secure login.
2. **Data Encryption**
   * AES-256 encryption for stored documents.
3. **Network Security**
   * Cloudflare DDoS protection and WAF implementation.
4. **Access Control**
   * Role-based access for applicants, administrators, and verification officers.

**6.0 API Documentation**

| **API Endpoint** | **Method** | **Description** |
| --- | --- | --- |
| /users/register | POST | Registers a new user. |
| /applications/submit | POST | Submits passport application. |
| /documents/upload | POST | Uploads identity verification documents. |
| /payments/process | POST | Handles passport fee payment. |
| /verification/status | GET | Retrieves police verification status. |

**7.0 Infrastructure Requirements**

| **Component** | **Technology** |
| --- | --- |
| Web Frontend | React.js |
| Mobile Frontend | React Native |
| Backend | Spring Boot (Java) |
| Database | MySQL |
| Cloud Storage | AWS S3 |
| API Gateway | Nginx |
| Security | OAuth 2.0, Cloudflare |

**8.0 Deployment Strategy**

* **Docker & Kubernetes**: Containerized deployment for scalability.
* **CI/CD Pipeline**: Automated testing & deployment via GitHub Actions.
* **Backup Strategy**: Daily database & document backup to AWS.

**9.0 Conclusion**

This document outlines the core **architecture, security measures, and system components** that will ensure the successful implementation of the **Passport Automation System**. Regular updates and reviews will ensure that the architecture remains aligned with business and security needs.

**10.0 Appendices**

* ER Diagram
* Data Flow Diagram (DFD)
* API Documentation
* Deployment Diagrams