**Security Manual for Schedule Advisement Software**

**1. Introduction**

**Purpose**: This manual provides detailed guidance on understanding the security mechanisms within the Schedule Advisement Software. It ensures that all team members are aware of the current security measures.

**2. Security Configurations**

**2.1 Encryption**

**BCrypt Password Encryption**:

* **Implementation**: We utilize **BCryptPasswordEncoder** for hashing user passwords, which provides a strong layer of security against password cracking techniques.
* **Configuration**: Defined in **BCryptEncoderConfig**, this bean setup ensures that **BCryptPasswordEncoder** is available application-wide.

**2.2 Authentication and Authorization**

**UserDetailsService Implementation**:

* **Purpose**: **UserDetailsServiceImpl** loads user-specific data for authentication. It is critical for verifying user credentials and enforcing security constraints based on user roles.
* **Details**: It checks if the user exists and whether their account is enabled, throwing **UsernameNotFoundException** if any conditions are unmet.

**DaoAuthenticationProvider**:

* **Setup**: Configured in **WebSecurityConfig** to use the custom **UserDetailsService** and **PasswordEncoder**. This provider facilitates authentication against the stored BCrypt hashes of user passwords.

**LoginSuccessHandler**:

* **Role**: Manages redirection post-login based on user roles and two-factor authentication status.
* **Details**: Directs users to specific pages based on their granted authorities and whether they have two-factor authentication enabled.

**2.3 Security Filter Chain**

**HttpSecurity Configurations**:

* **Function**: Defined in **WebSecurityConfig**, it specifies secured routes, configures login and logout mechanisms, and sets up exception handling.
* **Paths**: Uses **MvcRequestMatcher** and **AntPathRequestMatcher** to secure endpoints, ensuring that only authorized users can access specific functionalities.

**3. Two-Factor Authentication**

**Overview**: Adds an additional layer of security by requiring a second form of verification beyond just the username and password.

* **Process**:
  + **Generation**: **TwoFactorAuthentication** generates a temporary secret code sent to the user's email.
  + **Verification**: Users must enter this code upon login to gain access.

**4. Exception Handling**

**GlobalExceptionHandler**:

* **Function**: Provides a centralized exception handling mechanism across the application, crucial for handling validation errors and other runtime exceptions.
* **Benefits**: Improves API error responses by structuring them into understandable messages, enhancing the security by preventing exposure of sensitive error details.

**5. Resource Management**

**Static Resources Configuration**:

* **Implemented in**: **appConfig** class.
* **Purpose**: Securely serves static resources from predefined locations, avoiding unauthorized access and potential leakage of sensitive information.