Security

Least privileges [Mandatory]

Mobile applications should be designed with the least privileges on the device that it is installed on. For example, write access to the devices data store should not be sought unless it is essential for the mobile app to perform its functions.

Secure coding practices [Mandatory]

Follow secure coding best practices, such as input validation, parameterized queries, and output encoding, to prevent common vulnerabilities like SQL injection and cross-site scripting (XSS)

Multi-factor Authentication [Mandatory]

Multi-Factor Authentication (MFA) is strongly recommended as the primary authentication method for government institutions in Rwanda. It provides a high level of security by requiring users to present multiple independent factors for identity verification, significantly reducing the risk of unauthorized access

Session handling [Mandatory]

Session handling requires appropriate controls to be placed on the backend server to which the application connects. The backend server should treat the application as an untrusted entity; only allowing it access to content that it has been authorised to. When an application has authenticated, the backend server should enforce a session timeout, after which the application is forced to re-authenticate.

Sensitive data storage [Mandatory]

Sensitive information should not be stored on a device when it is not required. When sensitive data is required to be stored on a device, developers should look to make use of any native protected data storage APIs that are available to the platform. When it is no longer required on the device, it should be securely removed.

Encryption [Mandatory]

- Implement encryption algorithms to protect sensitive data both in transit and at rest. Use industry-standard encryption algorithms like AES (Advanced Encryption Standard) to secure user data
- Storing any cryptographic keys on the device will reduce the effectiveness of an additional cryptographic layer as keys stored locally could be recovered from the device (though these keys could be combined with a user credential to strengthen them). Storing the keys on a remote server would prevent an attacker with physical access to the device from retrieving them, though would require the application to authenticate to the server,

and have an internet connection.
- Encrypt sensitive data stored on the device, such as user credentials, personal information, and payment details. Apply encryption to local databases or use secure key storage mechanisms provided by the mobile operating system

Security audits [Mandatory]

Conduct regular security audits and penetration testing to identify vulnerabilities and address them promptly

Security updates [Mandatory]

Stay updated with the latest security patches and updates for the mobile operating system, libraries, and frameworks used in the app

---

Revision #3
Created 26 September 2025 13:22:44 by RISA
Updated 26 September 2025 13:32:04 by RISA