Test Plan

Moi is an application that aims to allow individuals to manage and share their identities in a secure, auditable way. Users of the app will be able to store personal information and identity documents and then grant access to parts of this information by generating a private key for the transaction and then providing the requesting party with the public key. All transactions will be stored in a blockchain, and so an unalterable history of facts will be available to all users.

Objectives

Our purposes in writing this test plan are to outline how we intend to verify that this application is both reliable and secure enough to be used by anyone and everyone for their identity management needs.

Scope

There are two main components of this application that will need to be tested, both individually and as a whole.

iOS Mobile App

The first component is the iOS mobile app. This will be the only user-facing part of this product, and so will need to be solid and reliable. The amount of trust users have in the app will be greatly influenced by the quality of the user interface.

Blockchain

The heart and soul of this product is the blockchain. As such it will need to be verified thoroughly. Specific considerations include ensuring blocks are added correctly, that consensus works as designed, and that individual transactions are secure and auditable in a way that enables privacy for users.

Testing Strategy

Unit and Integration Testing

Unit and integration testing will be automated and will be integrated into our build systems.

System Testing

We will need to verify that the mobile app can communicate correctly with the blockchain. We will also need to make sure that the nodes on the blockchain work together, and that the chain stays in sync as new blocks are added.

Security Testing

It will be important to test the security of the system from the app perspective as well as from the network and blockchain services perspective. All personal information must be kept securely encrypted, and all private keys must be generated securely and used correctly.

User Acceptance Testing

Testing with users will help us to verify that the problem has been solved, and that the experience is good.

Environment Requirements

iOS mobile phones equipped with the latest facial recognition software. We also need a blockchain network to be set up and active.

Test Schedule

Unit and integration testing will happen as units are being developed, so it will be happening continuously from 10/20 - 12/8. Major security testing will happen as individual high-risk tasks are completed. User acceptance and system testing will begin as soon as the minimum viable product has been completed in the first week of December.

Test Specification

1. ID Scanner
    1.1 Verify ID as driver license
    1.3 When non-driver license IDs are scanned, do not verify as license or capture data and display error to user
    1.2 Capture name, address, DOB, and driver license number from driver license
    1.3 Capture photo from driver license
2. Facial Recognition
    2.1 Verify that the individual's face was scanned correctly
    2.2 Verify that incorrect faces don't match, and display an error to the user
    2.3 Verify that the facial recognition module correctly matches the photo from the government issued id

2.4 Make sure faces from other people's ids don't match

3. Profile view
   - 3.1 Information displayed should be correct for the user
   - 3.2 Personal information should be well-protected on the device (password, pin, biometrics, etc.)
   - 3.3 Verify list of active keys is correct for the user
   - 3.4 Invalidating an active key should prevent it from being used
   - 3.5 Invalidating an active key should change the display for that key in the UI
   - 3.6 Clicking on an active key should show the personal information associated with that key
   - 3.7 Personal information not associated with a key should not be visible to those with whom the key is shared
4. Key Generation
   - 4.1 Generate unique private public key pair (one-time use keys)
   - 4.2 Generate available private public key pair (for public info)
   - 4.3 Verify keys are unhackable/unsolvable
   - 4.4 Verify unique public key private key pair does not ever generate twice
5. Key Verification
   - 5.1 Verify unique public key decrypts unique private key
   - 5.2 Verify unique private key decrypts unique public key
   - 5.3 Verify available public key decrypts available private key
   - 5.4 Verify available private key decrypts available public key
6. Blockchain
   - 6.1 Verify that blocks accurately store user data
   - 6.2 Ensure data is encrypted
   - 6.3 Ensure that blocks are distributed when added to the blockchain
   - 6.4 Verify blocks retrieve data from neighboring nodes
   - 6.5 Verify that rejected blocks are not added to blockchain
   - 6.6 Verify that data in blockchain is unhackable/unsolvable
7. Machine Learning
   - 6.1 Facial Recognition improvements
   - 6.2 Detect fraud