|  |
| --- |
| Testing Konzept Vault Guard |

Das Ziel dieses Testkonzepts ist es, eine fehlerfreie und benutzerfreundliche Nutzungserfahrung des Vault Guard Programmes zu gewährleisten. Das Konzept bestimmt die wesentlichen Bereiche, die überprüft werden müssen, um die reibungslose Funktionalität sicherzustellen.  
Wir möchten uns dabei primär auf integrierte / automatisierte Tests verlassen welche das refactoring vereinfachen sollten.

Den Fokus unseres Tests möchten wir auf die Grundfunktionen unseres Programmes konzentrieren aber nicht beschränken. So sollen die Funktionalitäten des LoginModels, des MainModel und die Models welche direkt mit den Entrys verknüpften sind ausgiebig getestet werden. Unsere Zusätzlichen Features welche nicht kritisch für das starten des Programmes sind werden mit weniger Tests bestückt. Dazu gehört der haveIBeenPawndService und die stärke Überprüfung eines generierten Passwortes. Die Controller werden ebenfalls vernachlässigt da sie nach Design keine grosse Logik beinhalten sollten und schwer zu testen sind aufgrund ihrer Interaktion mit den FXML Files.

Testarten:

|  |  |
| --- | --- |
| Unit - Tests | Durch Unit-Tests wird ein einzelner abgegrenzter Teil einer Applikation getestet, bspw. eine Methode. Dabei wird ein klarer Input- und Output-Wert definiert und verglichen. |

Testumgebung:

Wir nutzen IntelliJ IDEA als Haupt-IDE für die Entwicklung und Tests von Java-Anwendungen. Die Plattform ermöglicht effizientes Schreiben und Debuggen von zuverlässigen Tests. Dank IntelliJs JUnit5 Wizzard ist es möglich, Testvorlagen für alle zu testenden Methoden zu erstellen.

**Vault Guard Test-Klassen**:

Klasse: LoginModelTest

|  |  |  |
| --- | --- | --- |
| Test-Methoden | Objektzustand & Parameterwert | Erwartetes Ergebnis |
| testLogin() | Prüft die Korrektheit des Accountnamens und des Pwds. | Der aus dem File gelesene und entschlüsselte Wert ist korrekt |
| testLoginFailedFileReading() | Prüft die Unkorrektheit des Accountnamens und des Pwds. | File mit angegebenen Account Namen konnte nicht gefunden werden. |
| testLoginFailedDecrypting() | Prüft die Korrektheit des Accountnamens und die Unkorrektheit des Pwds. | File content mit angegebenen Pwd konnte nicht entschlüsselt werden. |
| testCreateNewUser() | Erstellt einen neuen User (ein neues File) | Überprüft den zugriff auf einige Methoden |
| testPropertyChangeEventLoginValid() | Testet ob der Listener bei einer Änderung korrekt informiert wird | Status Login wird zurückgegeben |
| testPropertyChangeEventNewUserValid() | Testet ob der Listener bei einer Änderung korrekt informiert wird | Status Login wird zurückgegeben |
| testPropertyChangeEventInvalid() | Testet ob der Listener bei einer Änderung korrekt informiert wird | Status Fail wird zurückgegeben |

Klasse: MainModel

|  |  |  |
| --- | --- | --- |
| Test-Methoden | Objektzustand & Parameterwert | Erwartetes Ergebnis |
| testSearchEntry() | Testet ob der Fuzzy search Algorithmus korrekt funktioniert | Testet dass vorgegebene Matches gefunden wurden |
| sortedEntryContentListTest() | Überprüft dass die vom MainModel verwaltete EntryList korrekt und sortiert übergeben wird | Eine alphabetisch nummerierte Liste wird Übergeben |
| testAddEntry() | Prüft dass ein Eintrag zur oben genannten Liste hinzugefügt werden kann. | Der Eintrag wird zur liste Hinzugefügt |
| testDeleteEntry() | Prüft dass ein Eintrag von oben genannten Liste entfernt werden kann. | Der Eintrag wird von der Liste entfernt |
| testSaveData() | Testet ob die in der Liste vorhandenen Daten encrypted und abgespeichert werden | Der Aufruf des Crypters und des Storagehandlers werden überprüft |

Klasse: PasswordTest

|  |  |  |
| --- | --- | --- |
| Test-Methoden | Objektzustand & Parameterwert | Erwartetes Ergebnis |
| testPasswordConstructorPositive() | Testet die korrekte Initialisierung des Constructors der der Passwort klasse | Vergleicht das mitgegebene Passwort und das gehashed Passwort |
| testPasswordConstructorNegativeNull() | Testet die korrekte Initialisierung des Constructors der der Passwort klasse | Testet auf das werfen einer exeption bei übergabewert Null |
| testPasswordConstructorNegativeEmpty() | Testet die korrekte Initialisierung des Constructors der der Passwort klasse | Vergleicht das mitgegebene leere Passwort und das gehashed Passwort |

Klasse: PasswordGeneratorTest

|  |  |  |
| --- | --- | --- |
| Test-Methoden | Objektzustand & Parameterwert | Erwartetes Ergebnis |
| testGeneratePasswordLength() | Testet auf die korrekte läge des generierten PWds | Die korrekte Länge des generierten PWds |
| testGeneratePasswordLengthTooShort() | Testet auf die unkorrekte läge des generierten PWds | Wirft eine exeption bei falscher (zu kurzer) länge |
| testVeryLongPassword() | Testet auf die unkorrekte läge des generierten PWds | Wirft eine exeption bei falscher (zu langer) länge |
| testIncludeNumerals() | Testet auf die korrektes format des Pwds | Überprüft, dass nummern enthalten sind |
| testIncludeUppercase() | Testet auf die korrektes format des Pwds | Überprüft, dass Grossbuchstaben enthalten sind |
| testIncludeSpecialSymbols() | Testet auf die korrektes format des Pwds | Überprüft, dass Sonderzeichen enthalten sind |
| testForbiddenCharactersExcluded() | Testet auf die unkorrektes format des Pwds | Überprüft, dass keine unerlaubten Zeichen enthalten sind |
| testRandomnessOverMultipleGenerations() | Testet die Zufälligkeit des generierten Pwd über mehrere Versuche (1000 Durchläufe) | Testet dass jedes generierte Pwd unique ist. |
| testEmptyCharacterSet() | Testet auf den Fall wenn kein Zeichen erlaubt werden sollte zur generierung eines Pwd | Überprüft Throw exeption |

Klasse: PasswordStrengthCalculatorTest

|  |  |  |
| --- | --- | --- |
| Test-Methoden | Objektzustand & Parameterwert | Erwartetes Ergebnis |
| testCalculateStrength () | Testet auf die korrekte encryptung | Passwort Strenge |
| testReadPasswordReferenceFiles () | Sie überprüft, ob die zurückgegebenen Listen von englischen Wörtern und bekannten Passwörtern nicht leer sind. | False wenn die Liste leer ist, true wenn die Liste nicht leer ist. |

Klasse: CryptographyHandlerTest

|  |  |  |
| --- | --- | --- |
| Test-Methoden | Objektzustand & Parameterwert | Erwartetes Ergebnis |
| testEncryptWithSameInputAndPassword() | Testet auf die korrekte encryptung | Testet auf die korrekte encryptung |
| testDecryptWithSameInputAndPassword() | Testet auf die korrekte decryptung | Testet auf die korrekte decryptung |
| testDecryptWithInvalidString() | Testet auf die unkorrekte decryptung | Wirft eine exeption bei falscher Parameterwerten |
| testEncryptDecryptLeadsToSameOutput() | Testet dass ein encrypten gefollgt von einem decypten wieder zu demselben output führt | Vergleicht Input mit Output nach beiden Operationen |

Klasse: StorageServiceTest

|  |  |  |
| --- | --- | --- |
| Test-Methoden | Objektzustand & Parameterwert | Erwartetes Ergebnis |
| testWriteWithValidInput() | Testet dass ein File korrekt geschrieben wird | Keine exeption wird geworfen |
| testWriteWithInvalidInput() | Testet dass ein File unkorrekt geschrieben wird | Eine exeption wird geworfen |
| testReadWithExistingFile() | Prüft dass schreiben und lesen eines Files | Gelesener Content entspricht dem gespeichertem Content |
| testReadWithNonExistingFile() | Testet dass ein nicht vorhandenes File gelesen wird | Rückgabewert ist Empty (nicht vorhanden) |
| testReadWithInvalidInput() | Testet dass ein falsch benanntes File gelesen wird | Eine exeption wird geworfen |

Klasse: HaveIBeenPawnedServiceTest

|  |  |  |
| --- | --- | --- |
| Test-Methoden | Objektzustand & Parameterwert | Erwartetes Ergebnis |
| testSHA1Hashing(String password) | Testet dass der angegebene Parameter korrekt gehashed wird | Testet dass der angegebene Parameter korrekt gehashed wird |
| testCheckBadPassword() | Testet ob ein schlechtes Pwd als geleackt erkannt wird | Wurde öfters als 0 mal enteckt |
| testCheckStrongPassword() | Testet ob ein gutes Pwd als sicher erkannt wird | Wurde noch nie enteckt |

Klasse: HashingTest

|  |  |  |
| --- | --- | --- |
| Test-Methoden | Objektzustand & Parameterwert | Erwartetes Ergebnis |
| testHashPassword() | Testet dass des Pwd korrekt gehashed wird | Testet dass des Pwd korrekt gehashed wird |
| testHashPasswordWithNull() | Testet ob Null gehashed wird | Wirft Exeption |
| testHashPasswordWithDifferentInputs() | Testet ob 2 unterschiedliche Passwörter unterschiedlich gehashed werden | Testet ob 2 unterschiedliche Passwörter unterschiedlich gehashed werden |
| testHashPasswordWithSameInput() | Testet ob 2 identische Passwörter gleich gehashed werden | Testet ob 2 identische Passwörter gleich gehashed werden |