**Sommario**

[**1. Test Email Validation** 1](#_Toc188183729)

[**2. Test Password Encryption** 2](#_Toc188183730)

[**3. Test Correct Hashing** 2](#_Toc188183731)

[**4. Test Character Position Finder** 2](#_Toc188183732)

[**5. Test Date-Time Formatter** 3](#_Toc188183733)

[**6. Test Default Profile Image Path** 3](#_Toc188183734)

[**7. Test Downloads Directory Path** 3](#_Toc188183735)

[**8. Test Downloads Folder Path** 4](#_Toc188183736)

[**9. Test Password Validation** 4](#_Toc188183737)

[**10. Test Note Types Loader** 4](#_Toc188183738)

**Functional Testing Document for Utils Class**

**Scope**

This document describes the functional testing process for the Utils class in the dev.uninotes.UniNotes.Utils package. Each test case is identified with a unique ID following the format TUTxx and includes details about the test scenario, expected outcomes, and actual results.

**Test Cases**

# **1. Test Email Validation**

* **Test ID**: TUT01
* **Objective**: Verify that the isValidEmail() method correctly identifies valid and invalid email formats.
* **Steps**:
  1. Call Utils.isValidEmail("test@example.com").
  2. Call Utils.isValidEmail("thisEmailIsInvalid").
* **Expected Result**: The method returns true for a valid email and false for an invalid email.
* **Successful**: yes.

# **2. Test Password Encryption**

* **Test ID**: TUT02
* **Objective**: Verify that the cryptPassword() method encrypts passwords correctly.
* **Steps**:
  1. Call Utils.cryptPassword("TestPassword123").
* **Expected Result**: The encrypted password is not null and is 32 characters long.
* **Successful**: yes.

# **3. Test Correct Hashing**

* **Test ID**: TUT03
* **Objective**: Verify that the cryptPassword() method encrypts the same.
* **Steps**:
  1. Call Utils.cryptPassword("TestPassword123").
  2. Call Utils.cryptPassword("TestPassword123").
* **Expected Result**: The encrypted passwords are equals.
* **Successful**: yes.

# **4. Test Character Position Finder**

* **Test ID**: TUT04
* **Objective**: Verify that the find() method correctly identifies the position of a character in a string.
* **Steps**:
  1. Call Utils.find("hello", 'l').
  2. Call Utils.find("hello", 'z').
* **Expected Result**: The method returns 2 for 'l' and -1 for 'z'.
* **Successful**: yes

# **5. Test Date-Time Formatter**

* **Test ID**: TUT05
* **Objective**: Verify that the dateTimeFormatter() method correctly formats a date-time string.
* **Steps**:
  1. Call Utils.dateTimeFormatter("2025-01-18 10:15:30").
* **Expected Result**: The method returns a LocalDateTime object with the correct year, month, and day.
* **Successful**: yes

# **6. Test Default Profile Image Path**

* **Test ID**: TUT06
* **Objective**: Verify that the getDefaultProfileImagePath() method returns the correct path.
* **Steps**:
  1. Call Utils.getDefaultProfileImagePath().
* **Expected Result**: The method returns "images/default/user.jpg".
* **Successful**: yes.

# **7. Test Downloads Directory Path**

* **Test ID**: TUT07
* **Objective**: Verify that the getDirectoryOfDowloads() method returns the correct downloads directory path.
* **Steps**:
  1. Call Utils.getDirectoryOfDowloads().
* **Expected Result**: The method returns "downloads/".
* **Successful**: yes.

# **8. Test Downloads Folder Path**

* **Test ID**: TUT08
* **Objective**: Verify that the getDownloadsFolder() method returns the correct downloads folder path.
* **Steps**:
  1. Call Utils.getDownloadsFolder().
* **Expected Result**: The method returns a path containing "Downloads".
* **Successful**: yes

# **9. Test Password Validation**

* **Test ID**: TUT09
* **Objective**: Verify that the validatePassword() method correctly identifies valid and invalid passwords.
* **Steps**:
  1. Call Utils.validatePassword("Test@123").
  2. Call Utils.validatePassword("test123").
* **Expected Result**: The method returns true for a valid password and false for an invalid password.
* **Successful**: yes.

# **10. Test Note Types Loader**

* **Test ID**: TUT10
* **Objective**: Verify that the loadNoteTypes() method retrieves a non-null and non-empty list of note types.
* **Steps**:
  1. Call Utils.loadNoteTypes().
* **Expected Result**: The method returns a non-null and non-empty list.
* **Successful**: yes.