# 22F-3733

**SQE-Assignment-3**

## Overview

SQE\_ASSIGNMENT-3 is a robust text editor application designed for creating, editing, and managing documents. It offers a user-friendly interface and a variety of features, including file creation, content editing, and Arabic text transliteration. Built with Java, the application utilizes a database for data persistence.

**JUnit Test Cases**  
To ensure the application functions correctly, JUnit tests will be implemented, focusing on unit testing the key features such as file operations, content editing, and transliteration functions. Each test case will verify the expected outcomes, including boundary conditions and error handling, to maintain high reliability and performance.

## Features

* Create, edit, and delete documents.
* Support for Arabic text transliteration.
* Retrieve and manage files from the database.
* User-friendly interface for easy navigation.

## Technology Stack

* **Language**: Java
* **Frameworks**: JavaFX for the UI
* **Database**: MariaDB for data storage
* **Testing**: JUnit for unit and integration testing
* **Build Tool**: Maven for project management

## Installation

**Prerequisites**

* **Java Development Kit (JDK)**: Version 8 or above
* **Maven**: Version 3.6 or above
* **MariaDB**: Installed and running

**Clone the Repository**

```

git clone https://github.com/AbbasHafeez/SQE\_ASSIGNMEN3-22F-3725.git

```

# Equivalence Class Partitioning (ECP) and Boundary Value Analysis (BVA)

## testSaveImportedFileToDatabase

### Equivalence Class Partitioning (ECP):

Valid Cases:

* - `content’: Non-empty string content (e.g., valid file content).
* - `fileName`: Valid filename string (non-empty and unique in the database).

Invalid Cases:

* - `content`: Empty string or null.
* - `fileName`: Empty string, null, or file name already existing in the database.

### Boundary Value Analysis (BVA):

* - Minimum content length: 1 character.
* - Typical content lengths, including large content approaching database column limit.
* - Boundary length for `fileName` (shortest valid name, longest allowed name by the system).

### Test Notes:

* - Check that content is saved correctly for valid inputs.
* - Verify handling of empty or duplicate filenames and content.
* - Ensure proper messaging for success or failure scenarios.

## 2.testSaveFile

### Equivalence Class Partitioning (ECP):

Valid Cases:

* - `fileName`: Valid, non-empty filename.
* - `contentArray`: Array with non-empty content.
* - `fileType`: Supported file type (e.g., 'txt').

Invalid Cases:

* - `fileName`: Empty or null.
* - `contentArray`: Null or empty content.
* - `fileType`: Unsupported file type.

### Boundary Value Analysis (BVA):

* - Minimum content length in `contentArray`.
* - Long filename approaching the system's maximum allowed length.
* - Various content sizes within `contentArray`.

### Test Notes:

* - Ensure successful file saving with valid data.
* - Confirm handling of invalid cases, including null or empty filename and unsupported file types.

## 3.testGetFileDetails

### Equivalence Class Partitioning (ECP):

Valid Cases:

* - `fileName`: Existing file in the database.

Invalid Cases:

* - `fileName`: Non-existent file or empty string.

### Boundary Value Analysis (BVA):

* - Minimum file name length.
* - Maximum file name length allowed by the system.

### Test Notes:

* - Confirm that file details are correctly returned for existing files.
* - Validate error handling or null response for non-existent or invalid file names.

## 4.testSaveContentWithPagination

### Equivalence Class Partitioning (ECP):

Valid Cases:

* - `textFileId`: Existing file ID in the database.
* - `content`: Non-empty string content.

Invalid Cases:

* - `textFileId`: Non-existent file ID.
* - `content`: Empty or null string.

### Boundary Value Analysis (BVA):

* - Minimum valid file ID.
* - Maximum content length that fits within pagination limits.
* - Minimum content length (1 character).

### Test Notes:

* - Ensure content with pagination is saved correctly for valid cases.
* - Validate handling of empty content and invalid `textFileId`.

## 5.testGetFileIdByName

### Equivalence Class Partitioning (ECP):

Valid Cases:

* - `fileName`: Existing file name in the database.

Invalid Cases:

* - `fileName`: Non-existent file name or empty string.

### Boundary Value Analysis (BVA):

* - Minimum file name length.
* - Maximum file name length allowed by the system.

### Test Notes:

* - Check that valid file names return a positive file ID.
* - Verify that non-existent or invalid file names return `-1`.