**PRACTICAL :-9**

**On the basis of required validations map the different test cases to handle all possible critical cases which may arise during the life cycle of the software .**

For a Unity-Based QR-Enabled Online Book Creator, there are specific considerations and critical cases that should be addressed through various types of testing. Here's a set of test cases mapped to handle critical scenarios throughout the software life cycle:

**Installation and Setup:**

Test Case 1: Installation Verification

* Verify that the software installs correctly on supported platforms (Windows, macOS, Linux).

Test Case 2: Dependency Check

* Ensure that all required dependencies (Unity version, QR code libraries) are installed and compatible.

**User Authentication:**

Test Case 3: Valid User Login

* Verify that registered users can log in successfully.

Test Case 4: Invalid User Login

* Check the system response to invalid login credentials.

Test Case 5: Account Lockout

* Test the system's response to multiple consecutive failed login attempts.

**QR Code Generation and Scanning:**

Test Case 6: QR Code Generation

* Ensure that QR codes are generated accurately for each book.

Test Case 7: QR Code Scanning

* Verify that the application can accurately scan QR codes associated with specific books.

**Book Creation and Editing:**

Test Case 8: Book Creation

* Ensure users can create a new book with the necessary details.

Test Case 9: Book Editing

* Verify that users can edit the content of an existing book.

**Data Validation:**

Test Case 10: Input Validation

* Validate that the system handles various inputs gracefully (special characters, long strings).

Test Case 11: Field Constraints

* Check for maximum and minimum character limits for book titles, descriptions, etc.

**Offline Mode:**

Test Case 12: Offline Book Creation

* Verify that users can create and edit books offline, with changes synced when online.

**Collaborative Editing:**

Test Case 13: Simultaneous Editing

* Test how the system handles multiple users editing the same book simultaneously.

**Error Handling:**

Test Case 14: Graceful Error Handling

* Check the system's response to unexpected errors, ensuring informative error messages are displayed.

**Performance:**

Test Case 15: QR Code Generation Speed

* Evaluate the time taken to generate QR codes for large books.

Test Case 16: Application Responsiveness

* Assess the application's responsiveness during heavy user load.

**Security:**

Test Case 17: QR Code Security

* Verify that QR codes are secure and not easily replicable.

Test Case 18: User Data Protection

* Ensure user data is encrypted and protected from unauthorized access.

**Usability:**

Test Case 19: User Interface Navigation

* Check the intuitiveness of the user interface for creating, editing, and managing books.

**Integration:**

Test Case 20: Unity Integration

* Ensure smooth integration with the Unity environment.

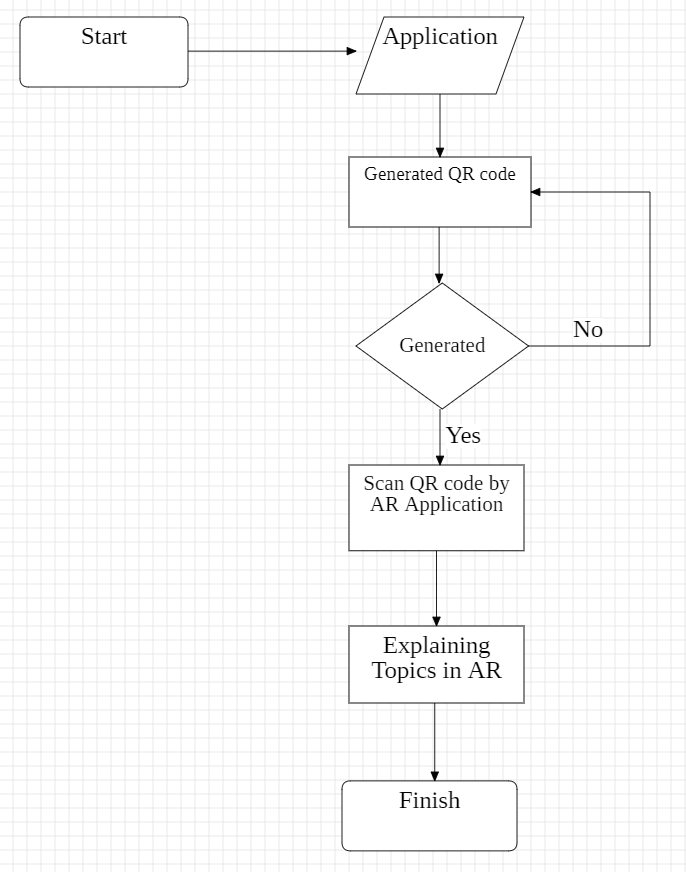
**Backup and Recovery:**

Test Case 21: Automatic Backup

* Verify that the system automatically backs up user data.

Test Case 22: Data Restoration

* Test the process of restoring data from a backup.



Test Case 25: Accessibility

* Evaluate the software for accessibility, ensuring it is usable by individuals with disabilities.
* Regularly update and expand the test suite as the software evolves through different phases of the development life cycle. Additionally, consider automation for repetitive and critical test scenarios to enhance the efficiency of your testing process