Deliverable 1 Test cases:

1. **Test Case: Successful Account Creation:**

This test case checks if the account is successfully created when valid data is provided.

@Test

public void testSuccessfulAccountCreation() {

// Setup

FirebaseAuth auth = FirebaseAuth.getInstance();

String email = "test\_user@test.com";

String password = "test\_password";

// Execute

auth.createUserWithEmailAndPassword(email, password)

.addOnCompleteListener(task -> {

// Verify

assertTrue(task.isSuccessful());

});

}

1. **Test Case: Account creation with Invalid data:**

This test case checks if the system correctly throws a validation error when trying to create an account with invalid data (in this case, an empty email).

@Test

public void testAccountCreationWithInvalidData() {

// Setup

FirebaseAuth auth = FirebaseAuth.getInstance();

String email = "";

String password = "test\_password";

// Execute

auth.createUserWithEmailAndPassword(email, password)

.addOnCompleteListener(task -> {

// Verify

assertFalse(task.isSuccessful());

assertTrue(task.getException() instanceof FirebaseAuthInvalidCredentialsException);

});

}

1. **Test Case: Account Creation With Existing Email:**

This test case checks if the system correctly throws an error when trying to create an account with an email that already exists.

@Test

public void testAccountCreationWithExistingEmail() {

// Setup

FirebaseAuth auth = FirebaseAuth.getInstance();

String email = "existing\_user@test.com";

String password = "test\_password";

// Execute

auth.createUserWithEmailAndPassword(email, password)

.addOnCompleteListener(task -> {

// Verify

assertFalse(task.isSuccessful());

assertTrue(task.getException() instanceof FirebaseAuthUserCollisionException);

});

}