Black box testing

Black box testing is a type of software testing in which the tester is unaware of the software's internal workings. Instead, the tester only interacts with the system's user interface, like input data and observes output generated by the system.

In our project academic affairs system, black box testing could involve testing various functionalities such as login and new user, student grade tracking, viewing course reports, etc. In this testing, we focused on verifying that the system behaves as expected based on its specifications and requirements.

1.Login (Student)

Fields:

- **email :**

  The system’s database should contain a username. So here we have cases that entered users who have

  1. email Match with system’s database (valid)
  2. emial Does not match with system’s database (invalid)

- **Password:**

  Entered password should match the respective user’s password in the database. So here we have two cases

  1. Password match with system’s database (valid)
  2. Password does not match with system’s database(invalid)

So for equivalence classes,

Username has two classes e1,e2;
Password has two classes p1,p2;

For equivalence partitioning, we will have 4 test case scenarios.

Equivalence	email	Password	Verdict
e1 p1	Match with database	Match with database	Valid login
e1 p2	Match with database	Not match with database	Invalid login
e2 p1	Not match with database	Match with database	invalid login
e2 p2	Not match with database	Not Match with database	Invalid login

2. Login (Faculty)

- **email :**
  The system's database should contain a username. From that, we have two cases :

  1. email Match with system's database (valid)
  2. email Does not match with system's database (invalid)

- **Password:**

  Entered password should match the respective user's password in the database.

  1. Password match with system's database (valid)
  2. Password does not match with system's database(invalid)

  So for equivalence classes,

  Username has two classes, e1 and e2;
  Password has two classes p1,p2;

For equivalence partitioning, we will have a total of 6 test-case scenarios.

Equivalence	email	Password	Verdict
e1 p1	Matched with database	Matched with database	Valid login
e1 p2	Matched with database	Not matched with database	Invalid login
e2 p1	Not matched with database	Matched with database	Invalid login
e2 p2	Not matched with database	Not matched with database	Invalid login

3. Login (Admin)

- **username :**

  Admin's username is hardcoded in the backend as ADMIN. So there are two cases:

  1. entered username is ADMIN (valid)
  2. entered username is not ADMIN (invalid)

- **Password:**

  Admin's password is hardcoded in the backend as PASSWORD. So there are two cases:

  1. Entered password is PASSWORD (valid)
  2. Entered password is not PASSWORD (invalid)

So for equivalence classes,

username has two classes, u1 and u2;
Password has two classes p1,p2;

For equivalence partitioning, we will have 4 test case scenarios.

Equivalence	username	Password	Verdict
u1 p1	ADMIN	PASSWORD	Valid login
u1 p2	ADMIN	Not PASSWORD	Invalid login
u2 p1	Not ADMIN	PASSWORD	Invalid login
u2 p2	Not ADMIN	Not PASSWORD	Invalid login

4. New User Registration (Student)

Fields:

- **email**

  This username should be inside the database. so there would be two cases

  1. email matched with database (valid)
  2. Email not matched with database (invalid)

- **Registration ID**

  Registration id should be matched with provided registration id by the university.so there would be two cases

  1. Registration id matched (valid)
  2. Registration is not matched (invalid)

- **Date of birth**

  Every possible date on the calendar could be the user's choice, except that any date would be wrong.

So for equivalence classes,

Username has two classes e1,e2
Registration id has two classes r1,r2
Date of birth has two classes d1,d2

So for equivalence partitioning, we have a total of 8 cases

Equivalence	email	Registration id	Date Of Birth	Verdict
e1 r1 d1	Matched with database	In the domain	Format Valid	Valid user
e1 r1 d2	Matched with database	In the domain	Format inValid	Invalid user
e1 r2 d1	Matched with database	Out of the do main	Format valid	Invalid user
e1 r2 d2	Matched with database	Out of the domain	Format Invalid	Invalid user
e2 r1 d1	Not matched with database	In the domain	Format Valid	Invalid user
e2 r1 d2	Not matched with database	In the domain	Format inValid	Invalid user
e2 r2 d1	Not matched with database	Out of the domain	Format valid	Invalid user
e2 r2 d2	Not matched with database	Out of the domain	Format Invalid	Invalid user

5. Broadcast Announcements (Admin)

Admin can broadcast any announcements to the faculty and students. Hence only one equivalence class is for a successful broadcast.

6. Course Reports retrieval (Faculty)

Faculty can view exam marks, quiz marks and assignment marks. Hence only one equivalence class is for a successful fetch of marks.

7. Maintaining Course information and Content (Faculty)

Fields:

- **Resource title**

  Any title would be valid, so two equivalent classes containing all possible titles should not be empty.

  1. Any possible title (valid)
  2. Empty title (invalid)

- **Link**

  Any link that refers to the respective document or article would be valid, so 1 equivalent class contains possible links.

  1. Should refer to the uploaded document (valid)
  2. Should not refer to an uploaded document (invalid)

- **Modify option**

  Modification could be done by adding a new document so one equivalent class for successful modification.

So equivalence classes,

The title has two classes for title t1,t2
The link has to refer to the respective document or article or video, or any other resource, so there would be two equivalence classes l1,l2,
Modification could be uploading any valid document type so there would be one equivalence class m1

So for equivalence partitioning, we have a total of 4 cases.

Equivalence	title	link	verdict
t1 l1 m1	Not empty title	Valid doc link	Successful modification

t1 l2 m1	Not empty title	Invalid doc link	Unsuccessful modification
t2 l1 m1	Empty title	Valid doc link	Unsuccessful modification
t2 l2 m1	Empty title	inValid doc link	Unsuccessful modification

8. Grade Tracking/Transcripts (Students)

Students can view their transcripts by going to the transcript option from the dashboard.

- **Choose Subject**

    1. Choose a valid subject (Valid)
    2. Choose a subject that the student has not taken any courses (Invalid)
    3. For a student who has not received any grades (invalid)

So for equivalence partitioning, we have a total of 3 cases.

Equivalence	Subject	Verdict
s1	Matched with database	Valid grades
s2	Matched with database	Invalid grades
s3	Matched with database but there is no update	Invalid grades

9. Course Reports - Individual Student

Each student can view his marks in a given exam, assignment or quiz, so one equivalent class for successful retrieval of marks.

10. Fees Receipt Report (Students)

Each student can view his fee payment details and download documents of his/her fee payment. So one equivalent class for successful fee receipt download.

11. Attendance Tracker (Student)

Each student can track their attendance in individual subjects by selecting their required subject. So one equivalent class.

12. Feedback System (Admin)

Admin can view the subject's feedback by choosing the subject.

- **Choose Subject**

    1. Choose a valid subject (Valid)
    2. Choose an invalid subject (Invalid)

So for equivalence partitioning, we have a total of 2 cases.

Equivalence	Subject	Verdict
s1	Matched with database	Valid feedback displayed
s2	Does not match with database	Invalid feedback displayed

13. Modify Student Information (Student)

Field :

- **Password**

  Any password would valid except empty password so there would be 2 equivalence classes.

1. Not empty password (valid)
2. Empty password (invalid)

So here would be two equivalence classes p1,p2

For equivalence table there would be two cases

Equivalence	Password	Verdict
p1	Not empty	Valid modification
p2	empty	Invalid modification

14. Modify faculty Information (Faculty)

Field :

- **Password**

  Any password would valid except empty password so there would be 2 equivalence classes.

  1. Not empty password (valid)
  2. Empty password (invalid)

So here would be two equivalence classes p1,p2

For equivalence table there would be two cases

Equivalence	Password	Verdict
p1	Not empty	Valid modification
p2	empty	Invalid modification