



# American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Summer 21 22

Section: B

Software Quality Assurance and Testing

**Dekho: Online Streaming Platform**

A Report submitted

By

SN	Student Name	Student ID
1	Sk. Sakib Islam	18-37593-1
2	Minhazul Hakim	18-39155-3
3	Refat, Mahfujar Rahman	19-40903-2
4	Showvik Salman Dhrubo	19-39801-1

Under the supervision of

**ABHIJIT BHOWMIK**

Associate Professor and Special Assistant [OSA]

Department of Computer Science

Faculty of Science & Technology

American International University-Bangladesh

---

# Software Test Plan

for

## Dheko: Online Streaming Platform

Version 1.0 approved

Prepared by

Showvik Salman Dhrubo

Minhazul Hakim

Refat, Mahfujar Rahman

Sk. Sakib Islam

18.08.22

## Checked By Industry Personnel

Name:

Designation:

Company:

Sign:

Date:

## Table of Contents

Revision History	3
1. TEST PLAN IDENTIFIER: RS-MTP01.3	3
2. REFERENCES	3
3. INTRODUCTION	3
Background to the Problem	3
Solution to the Problem	4
4. REQUIREMENT SPECIFICATION	4
4.1 System Features	4
4.2 System Quality Attributes	4
4.3 System Interface	5
4.4 Project Requirements	5
5. FEATURES NOT TO BE TESTED	5
6. TESTING APPROACH	5
6.1 Testing Levels	5
6.2 Test Tools	6
6.3 Meetings	6
7. TEST CASES/TEST ITEMS	6
8. ITEM PASS/FAIL CRITERIA	7
9. TEST DELIVERABLES	7
10. STAFFING AND TRAINING NEEDS	8
11. RESPONSIBILITIES	8
12. TESTING SCHEDULE	8
13. PLANNING RISKS AND CONTINGENCIES	9
14. APPROVALS	9

## Revision History

Revision	Date	Updated by	Update Comments
0.1	2022.08.15	Showvik Salman Dhrubo	First Draft
0.2	2022.08.16	Minhazul Hakim	Second Draft
0.3	2022.08.16	Minhazul Hakim	Third Draft
0.4	2022.08.16	Refat, Mahfujar Rahman	Fourth Draft
0.5	2022.08.17	Sk. Sakib Islam	Fifth Draft
0.6	2022..08.17	Refat, Mahfujar Rahman	Sixth Draft
0.7	2022.08.18	Showvik Salman Dhrubo	Seventh Draft

### 1. TEST PLAN IDENTIFIER: [RS-MTP01.3](#)

Requirement id	Date	Requirement Description	Dependencies	originator
OSLogin_1	16-08-2022	Sign in for the website	Valid user	Junior Developer
OSRegistration_02	16-08-2022	Valid user data	Check validation	Junior Developer
OSLogout_3	16-08-2022	Remove UI from website	Log out session	Junior Developer
OSChangePassword_04	16-08-2022	Verify user	User must be login	Junior Developer
OSShowPackage_05	16-08-2022	Package visualization	Availability of package	Junior Developer
OSDeletePackage_06	16-08-2022	Remove the package id	Package unique id	Junior Developer

## 2. REFERENCES

- o SRS Documents  
example: <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
- o Test Plan: <https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html>
- o Test Case: <https://www.guru99.com/test-case.html>

## 3. INTRODUCTION

### Background to the Problem

After a hectic day at the office or university or in our leisure time we always want to watch something for entertainment such as movies, series, or dramas. But we often use Torrent site which is illegal and very painful to download. There are some foreign OTT platforms out in the market but those don't have Bangladeshi content. And there are also some Bangladeshi ott platforms but those don't have a large amount of deshi content.

### Solution to the Problem

So, our goal is to make a platform where users can watch Bangladeshi content in one place without any trouble. Users can come to our system, buy subscriptions and watch their favorite movies or dramas easily. We will contact the owner of those content and make a deal that can help both the owner and the users. It is a really tough job to meet business objectives but if we once normalize this for content makers and users it will be much easier. There are some FTP Servers from different ISP but those are not accessible to everyone but our system will be accessible to everyone.

## 4. REQUIREMENTS SPECIFICATION

### 4.1 System Features

#### User/Customer Features

##### 1. System Registration

##### Functional Requirements

2.1 The software shall allow users to sign up with a username, password, email, date of birth, and address.

2.2 The password and email must be in a valid format.

Priority Level: High

Precondition: user has a valid user id and password

## **2. System Login**

Functional Requirements

2.1 The software shall allow users to log in with their given username and password

2.2 If the email and/or password has been inserted the wrong user must not be able to log in.

Priority Level: High

Precondition: user has a valid user id and password

## **3. System Logout**

Priority Level: High

Precondition: user must be logged in

## **4. Edit and Update profile**

Priority Level: High

Precondition: user must be logged in

## **6. Subscribe to Package**

Priority Level: High

Precondition: user must be logged in

## **7. Play/Stop/Next Video**

Priority Level: High

Precondition: user must be logged in

## **8. Add Watchlist/Save to watch later**

Priority Level: Medium

Precondition: user must be logged in

## **9. Add ratings and reviews**

Priority Level: Medium

Precondition: user must be logged in

# **Admin Features**

## **1 System Registration/Login/Logout**

Functional Requirements

2.1 The software shall allow the admin to sign up with a username, password, email, date of birth, and address.

2.2 The password and email must be in a valid format.

Priority Level: High

Precondition: admin has a valid user id and password

## **2 CRUD Users/Customers**

Priority Level: High

Precondition: user must be logged in

### **2.3 CRUD videos**

Priority Level: High

Precondition: admin must be logged in

### **2.4 CRUD Packages**

Priority Level: High

Precondition: admin must be logged in

### **2.5 CRUD Payment**

Priority Level: High

Precondition: admin must be logged in

### **2.6 CRUD Expenses**

Priority Level: High

Precondition: admin must be logged in

## **4.2 System Quality Attributes**

- 1. USABILITY:** The ease of usage can be used to measure this. The software must be simple to use. In an average of three minutes and as much as five minutes, one of our frequent users can publish about their product. Our system's functions are simple to master due to its straightforward navigation. The system is very simple to understand how to utilize for new or infrequent users.
- 2. RELIABILITY AND CORRECTNESS:** Every piece of software has bugs. Our program delivers the precise outcomes users are looking for, and its accuracy is promised. Because our program has been well tested and doesn't give any false results, it is more reliable for our users to use..
- 3. MODULARITY:** We must carefully create modules for every system. There are numerous components in our system, which have been merged to create a full system. Because we can quickly identify flaws in any module, our job as testers is made easier because we only need to focus on fixing that one module and don't need to consider other

modules. Additionally, because our system was created in a modular way, we can also add new features to it.

**4. MAINTAINABILITY:** This refers to how quickly the maintenance crew can do its work. Our maintenance team's primary responsibility is to patch bugs and update or add new features. All features can be altered with a development effort of 25 hours or less by our lone maintenance programmer.

**5. EFFICIENCY:** The system's primary positive feature. measured in terms of the amount of time needed to finish any task that has been given to the system. The program will work well on a broad basis.

**6. TESTABILITY:** It refers to how quickly the test team can complete their assignment. Since our system was designed in modules, testers will find it simple to use and they won't need to test every module individually to find and repair errors. Testability is very module dependent.

**7. FLEXIBILITY:** The work involved in changing a program so that it works. With no more than two hours of labor, a maintenance programmer who has been supporting this product for at least eight months should be able to deliver a copy of the updated output, complete with code changes and testing.

## 4.3 System Interface

**Login:**

[Dheko](#) [Home](#) [About](#) [Contact](#)

# Login

---

© 2022 - My ASP.NET Application



## Packages:

[Dheko](#) [Home](#) [About](#) [Contact](#)

### Packages

[Logout](#)  
[Create](#)

Monthly Plan

150 / TK

1 month free premium access. No ads. 2 Device

Suscribe

[Edit](#) [Delete](#)

Half Yearly Package

400 / TK

6 months free movies and series, No Ads

Suscribe

[Edit](#) [Delete](#)

Yearly Package

650 / TK

1 year free movies and series, No Ads

<https://localhost:44365/Home/About>

## Package Edit:

[Dheko](#) [Home](#) [About](#) [Contact](#)

### PackageEdit

**Name:**

**Price:**

**Desc :**

**Promo:**

---

© 2022 - My ASP.NET Application

## Package Create:

[Dheko](#) [Home](#) [About](#) [Contact](#)

# PackageCreate

Package

PName

PPrice

PDesc

PCode

Create

[Back to List](#)

© 2022 - My ASP.NET Application

## 4.4 Project Requirements

Effort Estimation:

Our project is to develop an application named “**Dekho: Online Streaming Platform**”.

Development Time = 3 Months

Required number of peoples = 4

Budget Estimation:

Duration in weeks =  $3 * 4 = 12$  weeks

Office days = 5 days

Working hours = 8 Hours

So Per week working hours is =  $(5 \times 12)$  hours

= 60 hours

So Total Working hours is =  $(40 \times 12)$  hours

= 480 hours.

Developer salary is = 1200 Taka

Total developers Salary =  $(1200 \times 480)$  Taka

= 576,000 Taka

<u>Expense</u>	Amount	Total Amount
Salary for 4 developers		576000 Taka
3 months office rent	$3 \times 15000$	45,000 Taka
Electricity and other costs		20000 Taka
1 month Maintenance	$4 \times 10 \times 1200$	48000 Taka

cost		
Travel Cost	1*5000	5000 Taka
Total Cost		694000 Taka
15% of the total cost(profit)		104100 Taka
Now the total budget is		798100 Taka

## 5. FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be tested in this release:

- There is a feature where Users can add a video to watchlists or save it to watch later that feature is not to be tested in this release of the software
- The user review and rating feature is also not to be tested due to low risk

## 6. TESTING APPROACH

**Unit Testing:** We will perform the unit tests first. We will test distinct software units or components in this test. Every class or method, as an illustration. The aim is to guarantee that every piece of software code performs as intended. Developers carry out these tests while an application is still in the development (code) stage. We will use the "White Box Testing" method in this phase.

**Integration Testing:** The integration will then happen in the second section. We will make sure that all software modules are correctly integrated and tested collectively during this testing. The many software modules that make up our product were created by four programmers. This level of testing aims to identify flaws in how several software modules interact when combined. We'll use the "Bottom-up Integration" method in this phase.

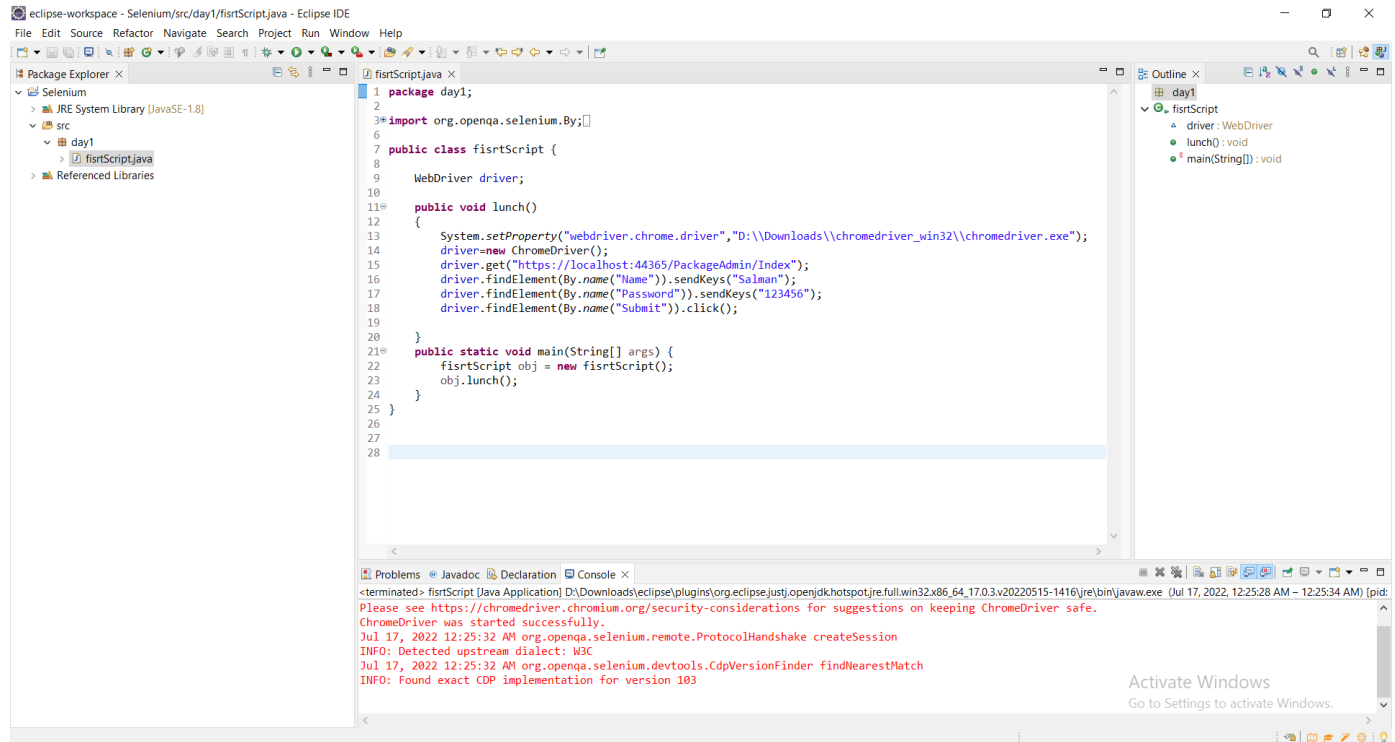
**System Testing:** After that, we'll test the system. We will evaluate a whole, integrated system using system tests. Then, we'll see if it satisfies all the criteria. Testing with a black box falls under this category. Therefore, we will use the "Black Box Testing" method at this level.

**Acceptance Testing:** Acceptance testing is the last phase of our testing. This test is carried out to ensure that the product is acceptable. To determine whether any mistakes were made during the functional testing stage, this test is conducted. We employ a "black box testing" strategy at this level. After that, we can rerun our unit tests.

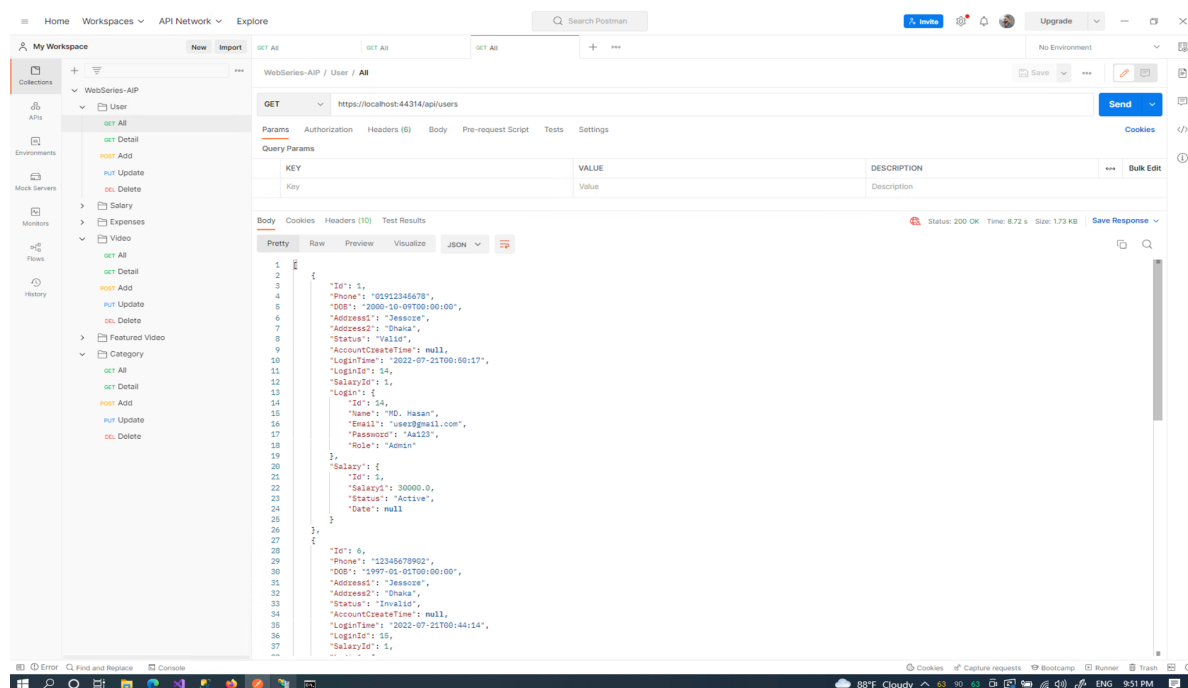
## 6.1 Testing Levels

- SYSTEM/INTEGRATION Testing will be performed by the test manager and development team leader with assistance from the individual developers as required. No specific test tools are available for this project. Programs will enter into the System/Integration test after all critical defects have been corrected. A program may have up to two Major defects as long as they do not impede testing of the program (I.E. there is a workaround for the error).
- UNIT TESTING will be done by the developer, and the results will be authorized by the team lead. Before unit testing may be accepted and passed on to the test person, the programmer must submit proof of unit testing (test case list, sample output, data printouts, defect information). The test person will receive all unit test information as well.
- The customer will do ACCEPTANCE testing with the help of the test manager and development team leader. After the System/Integration test is completed, the acceptance test will run in parallel with the existing manual ZIP/FAX process for one month.

### Selenium:



**Postman :**



Home Workspaces API Network Explore Search Postman

My Workspace New Import GET All GET All GET All GET Detail + ... No Environment

WebSeries-API / User / Detail

GET https://localhost:44314/api/user/6

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (10) Test Results Status: 200 OK Time: 886 ms Size: 773 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "Id": 6,
3   "Phone": "12345678902",
4   "DOB": "1997-05-01T00:00:00",
5   "Address1": "Jessore",
6   "Address2": "Dhaka",
7   "Status": "Invalid",
8   "AccountCreateTime": null,
9   "LoginTime": "2022-07-21T00:44:14",
10  "LoginId": 16,
11  "SalaryId": 1,
12  "Login": {
13    "Id": 16,
14    "Name": "Alvi Hasan",
15    "Email": "livealvi8@gmail.com",
16    "Password": "Aaz233",
17    "Role": "User"
18  },
19  "Salary": {
20    "Id": 1,
21    "Salary1": 30000.0,
22    "Status": "Active",
23    "Date": null
24  }
25 }
```

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search Web-Series-API-ASP.NET

Process: [16196] iisexpress.exe Debug Any CPU Continue

UserController.cs CategoryService.cs CategoryModel.cs Video.cs Category.cs

Web-Series-API-ASP.NET Web\_Series\_API-ASP.NET.Controllers.UserController Get()

```
26 Console.WriteLine(e);
27 return Request.CreateErrorResponse(HttpStatusCode.BadRequest, "Error no user found", e);
28 }
29
30
31 [Route("api/user/{id}")]
32 [HttpGet]
33 public HttpResponseMessage Get(int id)
34 {
35     try
36     {
37         return Request.CreateResponse(HttpStatusCode.OK, UserService.Get(id));
38     }
39     catch (Exception e)
40     {
41         Console.WriteLine(e);
42         return Request.CreateErrorResponse(HttpStatusCode.BadRequest, "Error user not found", e);
43     }
44 }
45
46 [Route("api/user/create")]
47 [HttpPost]
48 public HttpResponseMessage Post(UserModel user)
49 {
50     try
51     {
52         UserService.Create(user);
53         return Request.CreateResponse(HttpStatusCode.Created, "User added successfully");
54     }
55     catch (Exception e)
56     {
57         Console.WriteLine(e);
58         return Request.CreateErrorResponse(HttpStatusCode.BadRequest, "Error adding user", e);
59     }
60 }
61
62 [Route("api/user/edit")]
```

100% No issues found

Locals Search Name Value Type

Autos Locals Watch 1

Ready Web-Series-API-ASP.NET auth

Diagnostic Tools

Diagnostics session: 3:26 minutes

3:10min 3:20min

Events

Process Memory (MB)

CPU (% of all processors)

Summary Events Memory Usage CPU Usage

Events

Show Events (16 of 16)

Exceptions (0 of 0)

IntelliTrace Events (16 of 16)

Memory Usage

Take Snapshot

CPU Usage

Record CPU Profile

Error List Entire Solution Search Error List Code Description

## 6.3 Meetings

For a successful testing team, one of the main factors is to give the right job to the correct person. So holding regular team meetings is very important for a successful team. So the test team will meet once a week to evaluate each member's progress and determine whether or not they have completed their allocated tasks. And check if they are facing any issues. if they face any issues then other teams will support him and work to find a quick solution. The testing team leaders will meet with the project manager and development team once every two weeks. If there is any emergency there will be an extra meeting

## 7. TEST CASES/TEST ITEMS

### 7.1 Log in

Test the website login page by verifying valid username and password

Project Name: Dheko: Online Streaming Platform			Test Designed by: Refat, Mahfujar Rahman	
Test Case ID: OSLogin_1			Test Designed date: 16-08-2022	
Test Priority (Low, Medium, High): High			Test Executed by: Showvik Salman Dhrubo	
Module Name: Login Session			Test Execution date: 17-08-2022	
Test Title: verify login with a valid username and password				
Description: Test website login page				
Precondition (If any): User must have valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)



1. Go to the website 2. Enter username 3. Enter password 4. Click submit	Username: Salman  Password: 321	User should login into the application	As expected,	Pass
Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.				

## 7.2 Registration

Check the registration with a valid username, email, password, and date of birth

Project Name: Dheko: Online Streaming Platform		Test Designed by: Refat, Mahfujar Rahman		
Test Case ID: OSRegistration_02		Test Designed date:16-08-2022		
Test Priority (Low, Medium, High): Medium		Test Executed by:		
Module Name: Registration Session		Test Execution date:		
Test Title: Registration with a valid username, email, and password, date of birth, address				
Description: Test website registration page				
Precondition (If any): User must fill up all the input field.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
5. Go to the website 6. click the registration button 7. Enter all the valid information. 8. click “Submit” button	Username: Salman  gmail: <a href="mailto:Salman@gmail.com">Salman@gmail.com</a>  Date of birth: 16-01-1999  Gender: male  password: 123	User should register to the website.	As expected,	Pass

	address: Dhaka			
Post Condition: User is validated with database and successfully register an account.				

### 7.3 Log out

Test logout option is working or not

Project Name: Dheko: Online Streaming Platform		Test Designed by: Refat,Mahfujar Rahman		
Test Case ID: OSLogout_3		Test Designed date: 16-08-2022		
Test Priority (Low, Medium, High): High		Test Executed by:Showvik Salman Dhrubo		
Module Name: Logout Session		Test Execution date: 17-08-2022		
Test Title: verify logout option				
Description: Test website logout page				
Precondition (If any): User must be login				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
9. Go to the website 10. click logout		User should logout into the application	As expected,	Pass
Post Condition: User successfully logout to account. and go back to the home page				

### 7.4 Change Password

Check the change password feature is working accurately or not

Project Name: Dheko: Online Streaming Platform		Test Designed by: Refat,Mahfujar Rahman		
Test Case ID: OSChangePassword_04		Test Designed date: 16-08-2022		
Test Priority (Low, Medium, High): High		Test Executed by: Minhazul Hakim		
Module Name: Update password		Test Execution date: 17-08-2022		
Test Title: Can change password				
Description: Test password is update or not				
Precondition (If any): User must be login				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
11. Go to the website 12. login the site 13. write new password 14. click change password button	new Password: 123	successful update password	As expected,	Pass
Post Condition: update password is stored the database				

## 7.5 Show Package

check feature is showing or not

Project Name: Dheko: Online Streaming Platform		Test Designed by: Refat,Mahfujar Rahman		
Test Case ID: OSShowPackage_05		Test Designed date: 16-08-2022		
Test Priority (Low, Medium, High): Medium		Test Executed by: Minhazul Hakim		
Module Name: Show Package		Test Execution date: 17-08-2022		
Test Title: show all the package with their corresponding details				

Description: Test all the package are visible or not				
Precondition (If any): User Must have to login and go to package page				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
15. Go to the website 16. login the site 17. go package page		Can see all the package with details	As expected,	Pass
Post Condition:				

## 7.6 Delete Package

test case will check the test delete package option

Project Name: Dheko: Online Streaming Platform		Test Designed by: Refat,Mahfujar Rahman		
Test Case ID: OSDeletePackage_06		Test Designed date: 16-08-2022		
Test Priority (Low, Medium, High): Medium		Test Executed by: Sk. Sakib Islam		
Module Name: Delete package		Test Execution date: 17-08-2022		
Test Title: delete package with their unique id				
Description: Test the delete package option				
Precondition (If any): 1.User Must have to login and go to package page. 2.User must type the package id				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)

18. Go to the website 19. login the site 20. go package page  21. type the package id in the input field 22. click the delete option	Package id = "p1"	Remove the package from the package page	As expected,	Pass
Post Condition: the information that is stored in the database for the package will be removed.				

## 8. ITEM PASS/FAIL CRITERIA

We have used a total of 6 test cases in our implementation. Initially, 90% of the test cases that were applied to the system were passed successfully, while 10% of the test cases failed. Due to several database query-related problems, the test cases were failing. All test cases have been successfully completed when applied after a query-related issue has been resolved.

## 9. TEST DELIVERABLES

- o Acceptance test plan
- o System/Integration test plan
- o Unit test plans/turnover documentation
- o Screen prototypes
- o Report mock-ups
- o Defect/Incident reports and summaries
- o Test logs and turnover reports

## 10. STAFFING AND TRAINING NEEDS

This section outlines the hiring process and how to train employees for test positions. For the duration of the project, staffing is fixed. It's reasonable to presume that the majority of employees will consent to participate in some testing. The following professions are acknowledged:

**Project Manager:** Responsible for maturing the Web site's full execution. This entails developing requirements, coordinating the testing process, and managing the seller relationship.

**Test manager:** Fostering the expert test strategy, assessing the test deliverable, managing test cycles, gathering measurements and updating the project manager on progress, and advising when testing should be finished are all under their purview.

**Test Engineer:** This position's duties include organizing tests, developing test methods, developing test information, conducting tests, writing mechanized test strategies, making occurrence reports, reviewing episodes, and providing measures to the test administrator.

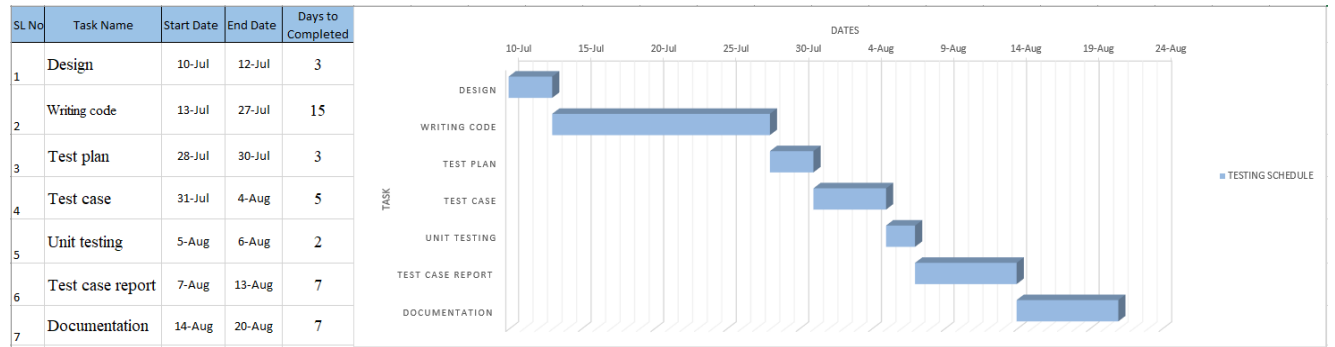
The website development life cycle should be known to the test manager and test professionals. This is only one example of the staffing and training needs because this project is being developed traditionally. The names of the conscious contributors to each initiative are so withheld.

## 11. RESPONSIBILITIES

Name	Role	Responsibilities
Hakim, Minhazul	Project Manager	<ol style="list-style-type: none"> <li>1. Completion of the Project.</li> <li>2. Execute all the test cases and report defects.</li> <li>3. Control whole project.</li> </ol>
Refat, Mahfujar Rahman	Quality Analyst	<ol style="list-style-type: none"> <li>1. Creation of test plans, test forms, test cases and test information.</li> <li>2. Carry out testing as per the characterized methods.</li> <li>3. Prepare all reports related to program testing carried out.</li> </ol>
Showvik Salman Dhrubo	Programmer	<ol style="list-style-type: none"> <li>1. Researching, designing, implementing, and managing software programs.</li> <li>2. Writing and implementing efficient code.</li> </ol>

		3. Deploying software tools, processes, and metrics.
Sk. Sakib Islam	Test Engineer	<p>1. Characterizing the testing activities All obligations of test planning</p> <p>2. To check in the event that the group has all the fundamental assets to execute the testing exercises.</p> <p>3. Prepare the report of testing activities.</p> <p>4. Overhauling extend directors frequently around the advance of testing exercises.</p> <p>5. Develop test cases and prioritize testing activities.</p> <p>6. Execute all the test cases and report defects.</p> <p>7. make documentation</p>

## 12. TESTING SCHEDULE



## 13. PLANNING RISKS AND CONTINGENCIES

Risk	Probability	Impact	Mitigation
Error in function	Medium	Medium	Regularly test the web application, and save a regular backup.
Give invalid input	High	High	Remind the user to enter the proper data type in each input area.
Loss of encrypted data(password)	Medium	High	Maintain security and backup



## 14. APPROVALS

Project Sponsor	Showvik Salman Dhrubo
Development Management	Refat, Mahfujar Rahman
EDI Project Manager	Minhazul Hakim
RS Test Manager	Sk. Sakib Islam
RS Development Team Manager	Showvik Salman Dhrubo
Reassigned Sales	Refat, Mahfujar Rahman
Order Entry EDI Team Manager	Minhazul Hakim