Kingdom of Saudi Arabia
Ministry of Education
Imam Abdulrahman bin Faisal University
Computer Science department
College of Science and Humanities

جامعة الإمام عبد الرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

CS 411 – Software Engineering
Term 1 – 2022/2023

Software Testing Plan
For

CSHJ

Version 1.0

CIS Year 4, G5

Dr. Norah Alnaim

30 October, 2022

This Software Testing Plan was prepared and provided as a deliverable for Software Engineering, CS 411, Term 1 and it will be used by the **CSHJ** team. This document is based in part on the IEEE Recommended Practice for Software Testing Plan.

Team Members

#	Student Name	ID
1	**May Mohammed AlOtaibi**	**2200004606**
2	**Reem Shaker Almualem**	**2190000429**
3	**Warood Khalid Alzayer**	**2190004986**
4	**Ghala Mohammed Alkhaldi**	**2200003157**
5	**Fida Mohammed Alelou**	**2200003041**

Table of Content

.

Table of Tables

Table of Figures

Revision History

Name	Date	Reason For Changes	Version
All members	Oct 26, 2022	Prepared initial version	0.1
All members	Oct 26, 2022	Updated section 3	0.2
All members	Oct 27, 2022	Updated section 4	0.3
All members	Oct 27, 2022	Updated section 5,6	0.4
All members	Oct 28, 2022	Updated section 7	0.5
All members	Oct 28, 2022	Updated section 8	0.6
All members	Oct 29, 2022	Updated section 9,10	0.7
All members	Oct 30, 2022	Complete review - Final version	1.0

1. Introduction

To get a perfect system and meet customer's requirements, you must go through a complete testing cycle. The software test plan is an essential part when creating a system to test it in every aspect. In this document we will describe the method applied to test the **CSHJ** platform. Including objectives, testing strategy, scope, reference, definition and acronyms.

1.1 Objectives

Software test plan (STP) document aims to give readers an overview of a project's testing strategy. There is a detailed description of the many features and specifications that will be evaluated as a part of the project's scope, the criteria for entering and leaving each phase, as well as their dependencies. Furthermore, it outlines the dangers and necessary countermeasures for testing and validating application's functionality independently.

1.2 Testing Strategy

The software test plan (STP) document used by the **CSHJ** platform to demonstrate the testing procedure. **Section 2** presents the test items included in the plan. **Section 3 and 4** presents the features to be tested and features not to be tested. **Section 5** contains a detailed description of test plan implementation. **Section 6** contains pass / fail criteria by specifying suspension criteria, resumption criteria and approval criteria. **Section 7** defines the methods and criteria used in performing test activities. **Section 8** Specify both the necessary and desired properties of the test environment including the physical characteristics, communications, mode of usage, and testing supplies. **Section 9** identifies the software test plan change management process, and **Section 10** presents the approval from all members for the plan.

1.3 Scope

All stages of testing will be covered in the Software Test Plan document. A variety of testing methods will be used depending on the type and element. Including Components testing, Integration testing, Conversion testing, Job stream testing, Interface testing, Security testing, Recovery testing, Performance testing, Regression testing, Acceptance testing and Beta testing.

1.4 Reference Material

[1] Job definition. [Online]. Available: https://www.ibm.com/docs/en/workloadscheduler/9.5.0?topic=objects-job-definition. [Accessed: 29-Nov-2021].

[2] "Definition of job stream," PCMAG. [Online]. Available: https://www.pcmag.com/encyclopedia/term/job-stream. [Accessed: 29-Nov-2021]

[3]Stallings, W. and Brown, L. (no date) *Computer security: Principles and practices.*

[4] Somerville, Ian, "Software Engineering", Addison Wesley (10thedition): 2016, ISBN:10:1-

292-09613-6, ISBN-13:978-1-292-09613-1

[5] Leung, H. K., & White, L. (1989, October). Insights into regression testing (software testing). In *Proceedings. Conference on Software Maintenance-1989* (pp. 60-69). IEEE.

[6] ["Overview of conversion testing in software testing," *GeeksforGeeks*, 29-Apr-2021. [Online]. Available: https://www.geeksforgeeks.org/overview-of-conversion-testing-in-software-testing/. [Accessed: 29-Nov-2021].

[7] "Data validation testing," Avenga, 22-Nov-2021. [Online]. Available: https://www.avenga.com/magazine/data-validation-testing/. [Accessed: 29-Nov-2021].

[8] IEEE Standard for Software Project. [Online] 1998. IEEE Std 1058-1998.

1.5 Definitions and Acronyms

Acronyms	***Definitions***
CSHJ	**College of Science and Humanities in Jubail**
STP	**Software Test Plan**
SRS	**Software Requirements Specification**
IEEE	**The Institute of Electrical and Electronic Engineering.**
IAU	**Imam Abdulrahman Bin Faisal University**
SDS	**Software Design Specification**
CS	**Computer Science**

Table 1 Acronyms and its definition

2. Test Items

This section will determine whether all of "**CSHJ's**" components—including their data, databases, and interfaces—adhere to the same GUI standards. While the SDS document will include details about the design and database, the SRS report will focus on the requirements.

2.1 Program Modules

All tests that will be completed by the developer and integrated into **CSHJ**. When a developer separates a program into small chunks, it is known as a program module. The program is designed as a unit however each component can be tested separately. Organizing and testing each system component's efficiency is made easier by dividing the software.

2.2 Job Control Procedures

The tester will start by evaluating each component independently, then move on to testing the functionality sequentially, then the interfaces, security, and performance, and eventually acceptability and test cases will be used to fix any errors or flaws that were missed during testing.

.

2.3 User Procedures

To verify that all user procedures implemented in **CSHJ** by users execute successfully, all of the processes will be tested. Section 5 of this report will detail the entire test.

2.4 Operator Procedures

To determine that **CSHJ** achieved user expectations, it performed various tests. the various types of testing handled at **CSHJ** :

- *Components testing*
- *Integration testing.*
- *Conversion testing.*
- *Job stream testing.*
- *Interface testing.*
- *Security testing.*
- *Recovery testing.*
- *Performance testing.*
- *Regression testing.*
- *Acceptance testing.*
- *Beta testing.*

3. Features To Be Tested

In this document (STD), the following functional requirements will be tested:

Common features:
- Sign-up to the system.
- Login to the system.
- Forgot password.
- Reset the password.

Administrations' features:
- Add and delete events.
- Change and add new rules.
- Edit club details.

Faculty members' features:
- Edit the attendance.
- Edit class schedule and times.
- Add a new notification.

Security members' features:
- Generate infractions.

Students' features:
- Review the attendance.
- Submit official excuses.
- Take notes.
- Raise objections.
- Enroll in a club.

4. Features Not To Be Tested

The table below shows the feature that will not be tested:

Features Not To Be Tested	Reasons
Make a group (Student)	**Not included in this release of the software.**
Download certificates	**Not included in this release of the software.**

Table 2 Features not to be tested

5. Approach

This section will provide the tests that have been performed in **CSHJ** application which are:

- Component testing
- Integration testing
- Conversion testing
- Job stream testing
- Interface testing
- Security testing
- Recovery testing
- Regression testing
- Performance testing
- Acceptance testing
- Beta testing

5.1 Component Testing

Component testing plays a crucial role in finding the problem. In order to ensure that each module of an application is working correctly and in accordance with requirements, component testing is always recommended prior to integration testing.

Technique: Test each component individually.

completion criteria: ensure that each component gives the expected result before moving to integration test.

5.1.1 Common function

5.1.1.1 Login

Test ID	**Login**
Prerequisite	**Have an account.**
Test procedure	**fill all required field then click login, the cases will appears:** **1- Correct email and password.**

	2- Correct email and incorrect password. **3- Incorrect email and correct password.** **4- Incorrect email and password**
Expected result	**1- Enter the platform successfully.** **2- An error message will appear "incorrect email or password".** **3- An error message will appear "incorrect email or password".** **4- An error message will appear "incorrect email or password".**
Actual result	**Based on the case, the expected result will be as stated.**
Verified (YES/NO)	**yes**

Table 3 Login

5.1.1.2 Forgot password

Test ID	**Forget password**
Prerequisite	**Have an account.**
Test procedure	**Enter username, email or phone number then click on forget password. Two important question will appear:** **1- Enter the new password** **2- Enter a confirmed password** **three cases will happen:** **1- The new password matches the old password.** **2- The new password matches the confirmed password.** **3- The new password doesn't match the confirmed password.**
Expected result	**1- An error message will appear "The new password is the same as the old password".** **2- A message will appear "your password has been updated successfully".** **3- An error message will appear "new password doesn't match confirmed password".**

Actual result	**Based on the case, the expected result will be as stated.**
Verified (YES/NO)	**yes**

Table 4 Forgot password

5.1.2 Admin

5.1.2.1 Add an event

Test ID	**Add an event**
Prerequisite	**Have an administrator account.**
Test procedure	**1- Click on plus sign (+) to add an event. 2- Fill the requirement (event name, date & time) 3- Determine who will appear. 4- Click post to share it**
Expected result	**1- The event information will appear to the selected people.**
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 5 Add an event

5.1.2.2 Delete an event

Test ID	**Delete an event**
Prerequisite	**Have an administrator account.**
Test procedure	**1- Click on dots sign (...) for the item that you want to delete. 2- Click on the delete. 3- A message will display "are you sure you want to delete this item?" 4- Click on "yes" 5- Click on "no"**
Expected result	**If the admin clicks on "yes" a message will appear "deleted successfully". If the admin clicks on "no" a message will appear "the event is not deleted".**
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 6 Delete an event

5.1.2.3 Add a club

Test ID	**Add a club**
Prerequisite	**Have an administrator account.**
Test procedure	**1- Click on plus sign (+) to add a new club. 2- Fill the requirement (club name, description, number of student) 3- Click add to add a new club.**
Expected result	**1- The new club will occur in the club main page.**
Actual result	**The expected result will be as stated.**

Verified (YES/NO)	**Yes**

Table 7 Add a club

5.1.2.4 Add a class

Test ID	**Add a class**
Prerequisite	**Have a faculty member account.**
Test procedure	**1- Click on plus sign (+) to add a class. 2- Fill the requirement (club name, description, number of student) 3- Click add to add a new club.**
Expected result	**1- The new class will occur in the class main page (faculty members can edit it).**
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 8 Add a class

5.1.2.5 Delete a class

Test ID	**Delete a class**
Prerequisite	**Have a faculty member account.**
Test procedure	**1- Click Edit for the item that you want to delete. 2- Click on the delete. 3- A message will display "are you sure you want to delete this item?" 4- Click on "yes" 5- Click on "no"**
Expected result	**If the admin clicks on "yes" a message will appear "deleted successfully".**

	If the admin clicks on “no” a message will appear “the class is not deleted”.
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 9 Delete a class

5.1.2.6 Add violations

Test ID	**Add violations**
Prerequisite	**Have a Security member account.**
Test procedure	**1- Go to the violation page. 2- Scan student barcodes to free a violation. 3- Fill the requirement (violation type, description) 4- Click add to add violation to the students.**
Expected result	**1- The violation will appear on the student page.**
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 10 Add violation

5.1.3 Student

5.1.3.1 Join student club

Test ID	**Join student club**
Prerequisite	**Have a student account.**

Test procedure	**1- Click on plus sign (+) to join any student club.**
Expected result	**1- Sending the request to join any student club to the specified club.**
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 11 Join student club

5.1.3.2 View attendance

Test ID	**View attendance**
Prerequisite	**Have a student account.**
Test procedure	**1- Go to the attendance page.** **2- Click on view.**
Expected result	**1- The attendance page will appear.** **2- View attendance in a specific course.**
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 12 View attendance

5.1.3.3 Upload absence excuse

Test ID	**Upload absence excuse**
Prerequisite	**Have a student account.**

Test procedure	**1- Go to the attendance page. 2- Click on upload absence excuse 3- Click on choose file**
Expected result	**1- The attendance page will appear. 2- You will upload your absence excuse. 3- All files will appear**
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 13 Upload absence excuse

5.1.3.4 View a violation

Test ID	**View a violation**
Prerequisite	**1-Have a student account. 2- Has a violation.**
Test procedure	**1- Go to the violation page. 2- Click on view past violation.**
Expected result	**1- The violation page will appear. 2- View all past violations.**
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 14 View a violation

5.1.3.5 Raise an objection

Test ID	**Raise an objection**
Prerequisite	**1-Have a student account. 2- Has a violation.**

Test procedure	**1- Go to the violation page.** **2- Click on raise an objection.**
Expected result	**1- The violation page will appear.** **2- Fill the requirement (violation ID, description). then click send to send it**
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 15 Raise an objection

5.2 Integration Testing

The integration testing is conducted once all the modules have been unit tested. It involves testing the interface between two software units or modules. During integration testing, faults in the interaction between integrated units will be exposed and an interface's correctness will be determined by this test.

.

Technique: The functionalities are tested incrementally to ensure their validity.

completion criteria: Assures that the entire system works as expected and interacts successfully. The test is done for the main user's interfaces, which are:

- Admin Homepage
- Student Homepage

5.2.1 Admin Homepage

Test ID	**Admin_Homepage**
Prerequisite	**Admin log in**
Test procedure	**1- Access "Edit event page" and do the component test cases to "add an event", "delete an event" as mentioned in (Section 5.1.2.1, 5.1.2.2)** **2- Access "Edit student clubs page" and do the component test cases to "add a club"as mentioned in (Section 5.1.2.3)**

	3- Access “Classes page” and do the component test cases to “add a class”, “delete a class” as mentioned in (Section 5.1.2.4, 5.1.2.5) **4- Access “Violation page” and do the component test cases to “add a violation” as mentioned in (Section 5.1.2.6)**
Expected result	**1- Same as the previous result that mentioned in (Section 5.1.2.1, 5.1.2.2)** **2- Same as the previous result that mentioned in (Section 5.1.2.3)** **3- Same as the previous result that mentioned in (Section 5.1.2.4, 5.1.2.5)** **4- Same as the previous result that mentioned in (Section 5.1.2.6)**
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 16 Admin homepage

5.2.2 Student Homepage

Test ID	**Student_Homepage**
Prerequisite	**Student log in**
Test procedure	**1- Access “View student clubs page ” and do the component test cases to “join student club” as mentioned in (Section 5.1.3.1)** **2- Access “Classes page” and do the component test cases to “View attendance”, “upload absence excuse” as mentioned in (Section 5.1.3.2, 5.1.3.3)** **3- Access “Violation page” and do the component test cases to “View a violation”, “Raise an objection” as mentioned in (Section 5.1.3.4, 5.1.3.5)**
Expected result	**1- Same as the previous result that mentioned in (Section 5.1.3.1)** **2- Same as the previous result that mentioned in (Section 5.1.3.2, 5.1.3.3)**

	3- Same as the previous result that mentioned in (Section 5.1.3.4, 5.1.3.5)
Actual result	**The expected result will be as stated.**
Verified (YES/NO)	**Yes**

Table 17 Student homepage

5.3 Conversion Testing

Since a software program relies on data to carry out its functions, each piece of software data is significant in software testing. Testing the conversion of a data format from one to another so that the software under test can use it continually is known as conversion. Data of any kind can be changed from one format to another, but web pages must be in HTML format for browsers to display them effectively. Using a tool or manually removing the data from the source database and re-inserting it in the destination database, also could convert data automatically.

Technique: Data validation test design.

Data validation testing ensures the information is accurate and comprehensive, and that the database and information can successfully undergo any necessary changes without suffering any losses. *Figure 1* illustrates the data validation testing strategies and procedures to aid in completing the aforementioned duties, in addition to having all the expected data accurately displayed according to all the inputs[3].

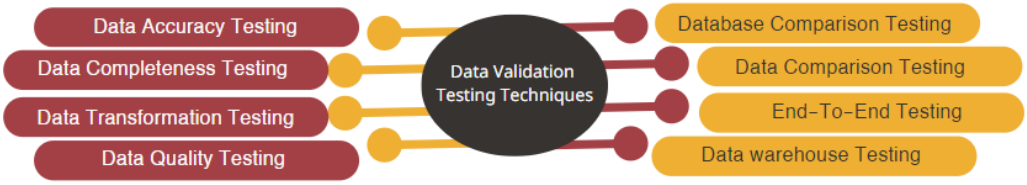

Figure 1 Data Validation Testing Techniques

Completion criteria: Ensuring that there is no alteration to the data elements or types during conversion from the old system format to the new system format.

5.4 Job Stream Testing

Function flow testing consists of a number of tasks that are completed concurrently and in a specific order dependent on one another's procedures. Each function will be unique in its

attributes and dependencies [1]. One function's outputs become the inputs for another, and so forth [2]. For instance, if the user wishes to alter their profile, they can select Edit their profile, make changes to their information (like their name or phone number), and then click Yes. Depending on one another, these actions will execute in a succession. Function flow testing aids in error detection and ensures that each task will be completed successfully.

Technique: Test stream of the functions.

Completion criteria: Ensure that each function corresponds to its purpose and achieves the desired result.

5.5 Interface Testing

The following **Table 18** shows interface testing to ensure that the flow of the system is as intended and meets the requirements and all interactions are works as they should be

Interface	User	Action
Login	All users	**1. Enter the ID and the password** **2. In case the ID and password are valid direct the user to the suitable homepage** **3. Else, an error message will show up**
Sign-up	All users	**1. Enter all required information and fill in all fields** **2. In case all information is valid and all fields have been filled in, a verification message will show up** **3. Else, an error message will show up.**
Forgot Password	All users	**1. Enter the email** **2. In case the email is found in the database, a link will be sent to the email** **3. Otherwise, an error message will show up.**
Join Club	Students	**1. If the club is full and the student is enrolled in another club, an error message will show up** **2. Otherwise, the student will be enrolled, and a message will appear.**

Generate Infraction	Security members	**1. The security member fills in all required information.** **2. In case all information is correct, a notification will be sent to the student.**
Add Exam Schedule	Faculty member and admins	**1. Enter all information and fill in all required information.** **2. In case the exam schedule is already added, an error message will appear** **3. Otherwise, a message will appear, and the exam schedule will be added.**
Add Event Schedule	Admins	**1. Enter all information and fill in all required information.** **2. In case the event schedule is already added, an error message will appear** **3. Otherwise, a message will appear, and the event schedule will be added.**

Table 18 Interface testing

5.6 Security Testing

The goal of the security testing is to ensure that all data and information of the users and system are secured. The systems must guarantee that only authorized users can access the system and it must be at a very high level of security to meet the stakeholders' expectation. the following are example of security performance:

- ➔ checking and verifying the system (**CSHJ**) is protected against unauthorized access.
- ➔ checking and verifying the system (**CSHJ**) transactions with the database are protected against network eavesdropping[3].
- ➔ verifying the password when it is changed through a link that is sent to the users
- ➔ verifying the users through the face ID.

Technique: only users who have an academic ID that is provided from the IAU university can join the system. And only valid passwords are accepted.

Completion criteria: only authenticated and authorized members in the IAU can access the **CSHJ** system.

5.7 Recovery Testing

In software engineering, Recovery Testing is a type of Non-Functional Testing. Testing for recovery ability is one of the methods used to validate software's ability to recover from failures like software or hardware crashes, network outages, etc. After a disaster or integrity loss, Recovery Testing determines whether software operations can be resumed. As part of recovery

testing, software can be returned to a known state of integrity and transactions can be reprocessed to the point at which they failed.[4]

Technique: Backup and follow the Recovery Process Life Cycle. As shown in Figure 2, the recovery process can be divided into five steps.[2]

1. Normal operation: A system is made up of hardware, software, and firmware that work together to accomplish a single goal. The system is expected to complete the job without interruptions within a specified period of time.
2. Disaster occurrence: Disruption may occur due to malfunction of the software due to various reasons like input-initiated malfunction, software crashing due to hardware failure, or damage due to fire, theft, and strike.
3. Disruption and failure of the operation: Disruptions invariably lead to loss of business opportunities, relationships, person-hours, and financial losses. In order to minimize disruption during the recovery phase, every agency should have a disaster recovery plan.
4. Recovery process: Recovery can be achieved without much loss of time, effort, and energy if a backup plan and risk mitigation processes are in place before disasters occur. To fix responsibility and help the organization avoid long disruption periods, a designated individual, along with his team, should be defined, along with their assigned roles.
5. Reconstruction - Normal process: Bringing the entire system back to regular operation by reconstructing all processes and information. All folders, as well as configuration files, may need to be rebuilt in multiple sessions. It is essential to have proper documentation and reconstruction processes to ensure a successful recovery.

Completion criteria: The following tests can be made to guarantee that files are correctly recovered once folders and files have been restored:

- Comparing folders and files.
- Renaming the damaged document folder is advised.
- Counting the number of files in the restored folders with a folder that already exists.
- Opening the files to see if they can be accessed.
- Check to see whether the data can be changed or accessed.

Figure 2 Recovery Process Life Cycle.[2]

5.8 Performance Testing

The performance test must ensure having the minimum response time which measures the speed of the application. Also, it ensures availability, portability and scalability. For response time, it must be as minimum as possible with respect to the load conditions that might happen, so the end users don't have to wait for a long time to process any action. Furthermore, the application must be always available and accessed by the users. In addition, ensure that the platform runs in all operating systems such as IOS and Android. Moreover, the application must be scalable due to the expected growth in the future.

Some performance tests implemented are:

- Check the maximum number of users that the application can handle before it crashes.
- Check if the application works in different platforms such as IOS and Android, then compare the speed of these platforms.
- Check the maximum number of transactions could be done in a second.
- Check the application speed in different internet connections.
- Check the application speed in uploading pictures/video/voice.
- Check if the application backs up the database successfully every day.

Technique: Testing in different situations and ways to ensure that the response time, availability, portability and scalability are performed as expected.

Completion criteria: All the test cases are completed within an acceptable time.

5.9 Regression Testing

Software testing includes regression testing. Re-running test cases ensures that the application's previous functionality continues to perform as intended and that the recent changes did not introduce any defects.

Techniques: Depending on specification changes, two types of regression tests can be identified and used [2]. These are:

1. Progressive regression testing:In order to reflect the addition of new features to the software, the specification will be modified. It is necessary to use the modified specification for progressive regression testing. The regression testing process will usually include testing modified software against a modified specification since new modules will be added to the software system.
2. Corrective regression testing: Specifications remain unchanged during corrective regression testing. In this case, only a few instructions in the software and perhaps some design decisions have been changed. Since the previous test plan specifies the input-output relationship correctly, most of the test cases in the previous test plan are considered valid. Some existing test cases may no longer test the previously targeted program constructs due to possible changes in the software's control and data flow structures. After the software has been corrected, corrective regression testing is frequently performed.

Completion criteria: To ensure that the applied changes to the application have not adversely affected previously tested functionality.

5.10 Acceptance Testing

In acceptance testing, the software must meet the stakeholder requirements so before releasing the system, it must be delivered to the stakeholders to ensure that it meets their requirements. The acceptance testing is considered to be one of the final stages of the software testing [4].

Technique: Take feedback on the software to make sure it works as it should be.

Completion criteria: The **CSHJ** software will be released if it is error free.

5.11 Beta Testing

Beta testing, done by the customer, using a pre-release version of the product to verify and validate that the system meets business functional requirements. The purpose of beta testing is to detect application faults, failures, and defects. **CSHJ** works with a company that has specialist people in testing the entire system. After taking the feedback, **CSHJ** developers should consider any error, failure and change that might occur, then revise and modify it to get a correct and smooth system.

Technique: Test the system by the users, get their feedback then the developers revise and fix the problems.

Completion Criteria: Test the entire system and ensure it works as expected.

6. Pass / Fail Criteria

This section includes suspension criteria, resumption criteria and approval criteria.

6.1 Suspension Criteria

As long as the following criteria are met, the test results will be accepted:

- There are no requirements available.
- System glitches are present.

6.2 Resumption Criteria

Testing may be resumed if the following requirements are met:

- The requirement is back in service.
- The system glitch has been resolved.

6.3 Approval Criteria

Test results will be accepted as long as the following criteria are met:

- Low risk.
- Achieves the desired result.
- It will not affect hardware or software.

7. Testing Process

In this section, we will determine the test methodology and criteria that **CSHJ** uses for performing test activities as well as subsections for test deliverables, test tasks, responsibility, resources, and schedule.

7.1 Test Deliverables

In the **CSHJ** project, the Software Test Plan (STP) will be created due to the testing process. After the testing process is complete, **CSHJ** will be ready for delivery.

7.2 Testing Tasks

The test part is the most significant part of every software project. Inaccurate project planning can significantly damage the running of tests and the quality of the platform. Thus, the task should break down carefully and wisely. Testing tasks should consist of the following:

Prepare the SRS and SDS documents

Arrange STP document
Prepare the hardware test environment.
Prepare the software test environment.
Set up each team member's tasks.
Implement all test activities using a different approach.
Manage errors encountered during testing.
Save the implementation in the event of a modification or update.
Maintain the application in the event of a change.

Table 19 Student homepageTesting Tasks

7.3 Responsibilities

All team members are responsible for testing the platform within the constraints attributes of each component and all testing tasks. This includes the preparation, management, and design of the platform and document and the handling of bugs during the test phase.

7.4 Resources

The team members will use the resources shown in **Table 20** to complete the system testing phase.

Resource	Description
Human	Skills and knowledge are important to all members of the team.
Hardware	An internet-connected PC with a high-speed internet connection.
Software	• NetBeans 8.2 • MySQL Workbench • HTML • PHP

	• CSS
Network	A network simulates the user's environment using LANs and internet connections
Test Tool	The development of a tool that will automatically generate test results in a predefined format, along with the automated execution of tests

Table 20 Resources

7.5 Schedule

Each testing task is listed in **Table 21** along with its estimated date.

Task	Date
Create test specification	18/10/2022
Perform test execution	20/10/2022
Error logs and execution logs	22/10/2022
Test Results/reports	25/10/2022
Modify the system	27/10/2022
Modify Software testing plan (STP)	28/10/2022
Test delivery	30/10/2022

Table 21 Schedule

8. Environmental Requirements

This section contains the hardware, software, security, tools, publications, risks and assumptions related to the application.

8.1 Hardware

The following hardware is required to test **CSHJ** application:

- High-speed internet connection
- Personal computer

8.2 Software

The following software is required to test **CSHJ** application:

- NetBeans 8.2
- MySQL Workbench
- HTML
- PHP
- CSS

8.3 Security

To ensure that the system have high security, these steps are applied:

- Use reliable resources to develop the system.
- Each user must use a unique username and difficult password at least 8 characters long to enter the system.
- If the password is wrong, the user will not be able to enter the system.
- When the user forgets the password a link will be sent to its registered email to restore the password.
- Facial recognition technology will be applied to minimize the entrance of unauthorized access.

.

8.4 Tools

The **CSHJ** application uses tools to test the website and check its validation and verification. The following tools are required to carry out the testing of **CSHJ** application activities:

- TestComplete
- TestingWhiz
- Katalon studio
- HERA
- TestRail
- OTP

8.5 Publications

The following documents are required to support the software test plan (STP):

- Software Requirement Specification (SRS).
- Software Design Specification (SDS).

8.6 Risks and Assumptions

In order to get the software to meet the customer requirements, the software must go through all

testing stages. Also, during these stages it may go through other stages. For example, any new requirements from the customer, new technology or competitors. These stages may put the system in a risk position. The risks and their contingency plan will be discussed in the table below:

Risks	Explanation	Contingency Plan
Test item availability	If the required item of testing isn't available. This will stop the testing phase and all the work will be delayed.	Using another available item to test until the wanted item is available.
Test recourse availability	If the required recourse of testing isn't available. This will stop the testing phase and all the work will be delayed until it becomes available.	Try to find another resource that does the same purpose.
Time constraints	Lose control of the time. Some tasks take longer than is scheduled or due to any new changes.	Creating a plan B for these situations. For example, create another schedule at the beginning of the work for contingency situations.

Table 22 Risks and Assumptions

9. Change Management Procedures

CSHJ procedures might be changed based on the team members. To obtain permission, a series of steps must be followed. Determine the problems that can be improved upon first. Second, go over the issue and potential fixes with the test manager. Third, hold a meeting with all project participants, the project manager, and stakeholders to discuss and solicit suggestions for additional adjustments. The plan will be amended to reflect the requested change if all participants approve of it. and the modification must be recorded. The client is free to make any changes to the client's shop pages that correspond to the SDS report. If there are any suggestions, they should be based on the needs. The customer could communicate with **CSHJ** personnel to pass along these suggestions to project members

10. Plan Approvals

Name	Signature	Date
Dr. Norah M. Al-naim		**Oct 30, 2022**
May Mohammed AlOtaibi		**Oct 30, 2022**
Reem Shaker Almualem		**Oct 30, 2022**
Warood Khalid Alzayer		**Oct 30, 2022**
Ghala Mohammed Alkhaldi		**Oct 30, 2022**
Fida Mohammed Alelou		**Oct 30, 2022**

Table 23 Plan Approvals