System Test Plan

for

Time-Table Generator

Version 1.0

CS-08

Indian Institute of Information Technology Vadodara

Team Members

Aman Yadav (201651007)

DakshKumar Gondaliya (201651014)

Kirtika Singhal (201651024)

Mayank Pathela (201651029)

Nikhil Sachan (201651034)

Parmeshwar Kumawat (201651035)

Revision History

Version	Date	Name	Description
1	02/10/2018	Parmeshwar Kumawat	Initial Document

1. Purpose

This document is the system test plan for the Time-Table Generator application. It describes the testing procedure which the testing team will use to evaluate if the application meets the established requirements of the client before the release. It keeps track of possible tests that the application should pass when development is completed.

2. Objective

The product supports the intended business functions and achieves the required software standards.

3. Test Items

The software products that need to be tested are:

- User Interface
- TimeTable generating algorithm
- Storing of data in the database
- Fetching of data from the database
- API calls

4. Functions that need to be tested

This is a listing of what is to be tested from the USERS viewpoint of what the system does. The Modules in the scope of testing for TimeTable Generator are as follows:

- User Registration
- User Log-In
- User Logout
- Set-up and edit profile
- Delete User Account
- Add Subject, Labs and Electives
- Delete Subject, Labs and Electives
- Add Teachers
- Delete Teachers
- Add ClassSection
- Delete ClassSection
- Add Slots

- Delete Slots
- Generate Time-Table

5. Overall Approach and Strategy

5.1. Testing Strategy

The Time-Table Generated Testing will include testing of all functionalities that are Identified in the scope. System testing activities will include the testing of new functionalities, Modified functionalities, screen level validations, work flows, functionality access, testing of internal-external interfaces.

5.2. Testing Types

5.2.1 Usability Testing

User Interface(UI)

Test Scenario	Test Steps	Expected Output
UI designing is user friendly		Application should be user friendly enough to display its content properly.

5.2.2 Functional Testing

A. Login

- **Precondition:** User has existing valid username and password.
- **Postcondition:** User is validated with the entries in the database and successfully logged in to the account.

Test Scenario	Test Steps	Expected Output
Login with valid username and password	Login to application and validate username and password	User should be able to login
Check that user does not leave any field blank.	User must be informed about the fields left blank.	User is not allowed to login without entering email or password.

B. Logout

- **Precondition:** User is Logged in.
- **Postcondition:** User is Logged out.

Test Scenario	Test Steps	Expected Output
Check for successful logout	Click on logout button from any page	User should be logged out and should return to the home page.

C. Input Data

- **Precondition:** User is Logged in.
- **Postcondition:** Input should be shown after data is inserted.

Test Scenario	Test Steps	Expected Output
Blank input is given	Click Add Input Button	Error Message
Redundant Data is Given in case of input of Teacher, Subject, Class and Section, and slots	Given Already Existing Data	Error Message
Input Valid Data	Click Add Input Button	Input is added Successfully

D. Delete Data and Account

- **Precondition:** User should be logged in.
- **Postcondition:** Data Should be deleted and user should be redirected to home page in case of Account deletion.

Test Scenario	**Test Steps**	**Expected Output**
Choose a Data to be deleted	Click Delete Data	Data Should be Deleted
Go to Profile and Click delete my account	Try to login with same email id	Error Message: User ID not exist

E. Generate Time-Table

- **Precondition:** User should be logged in.
- **Postcondition:** Time-Table is generated and can be view from profile section anytime..

Test Scenario	**Test Steps**	**Expected Output**
Input Slots	Click on Generate Time-Table	Time-Table page is shown with valid output.

6. Suspension Criteria and Resumption Requirements

6.1. Suspension Criteria

Testing will be suspended if the incidents found will not allow further testing of the system/application under-test. If testing is halted, and changes are made to the software or database, it is up to the Testing Manager to determine whether the test plan will be re-executed or part of the plan will be re-executed.

6.2. Resumption Requirements

Resumption of testing will be possible when the functionality that caused the suspension of testing has been re-tested successfully.

6.3. Test Data

Test data requirements are drawn up based on the functional requirements that are due for testing. The testing team will identify test cases that can be grouped into test scenarios and detail the data required to complete the testing activities.

7. Execution Plan

The execution plan will detail the test cases to be executed. The Execution plan will be put together to ensure that all the requirements are covered. The execution plan will be designed to accommodate some changes if necessary, if testing is incomplete on any day. All the test cases of the projects under test in this release are arranged in a logical order depending upon their interdependence.

8. Assumptions

• Production like data required will be available in the system prior to start of Functional Testing.
• All the requirements including system requirements and data requirements will be completed prior to testing.

9. Risks and Contingencies

• Lack of availability of required data, software or tools.
• Changes to the original requirements or designs.