**Name: Simran Dhoju**

Sprint Test Plan

[**Test plan Identifiers**](#_4zojg6uoob25) **2**

[**Reference**](#_l7cd9pqzg7le) **2**

[**Introduction**](#_eiol25aaqs6t) **2**

[**Purpose**](#_eraw2v5hfjde) **2**

[**Project Overview**](#_y1nv2vat68c1) **3**

[**In Scope**](#_3zwyrvv3zdlz) **3**

[**Out of scope**](#_20j8vuubxvr0) **4**

[**Testing Strategy/Approach**](#_obvx0zd6i72v) **4**

[**Test Item**](#_2wtkv98oggel) **5**

[**Tools**](#_fe9up7qa722y) **5**

[**Assumption / Risks**](#_2qleikuhu1fr) **6**

[ASSUMPTION](#_h1balpuwjadx) 6

[RISKS](#_d0r7nbw0arjj) 6

[**Test Deliverables**](#_pskn7whwc8p0) **7**

[**Testing Environment**](#_3dkcu2z9d8op) **7**

[**Bug Report**](#_hwjpvdz7u0zt) **7**

# Test plan Identifiers

Master test plan for School Management System software v2.0

# Reference

- Mockups ( Invision app).

- Requirement document.

# Introduction

The Test Plan is designed to prescribe the scope, approach, resources, and schedule of all testing activities of the project “School Management System”.

The plan identify the items to be tested, the features to be tested, the types of testing to be performed, testing strategies, testing environment, and the risks associated with the plan.

# Purpose

The purpose of the test plan is to validate all the requirements provided by the client has been fulfilled or not. The document will also describe the activities related to analyzing and testing the data being loaded in various entities are correct.The document will also talk about scope, limitations and priorities. The main purpose of this test plan are :

* To analyze the product
* To design the test Strategy
* To define the test Objectives
* To define test criteria
* To resource planning
* To plan test environment
* To schedule & estimation
* To determine test deliverables

# Project Overview

School Management System is carried on by any individual or institution engaged in providing services to school administrator to perform one or more of the following functionalities: Student View List(Name, Student\_id, ACT, SAT), Student Detail

View(individual detail of student), View scores of individual student, Search student according to name, score, Sort student scores, Export Student details in CSV format, etc.

The application has a login page, if provided with valid credential it will go to the main dashboard.

The main dashboard consists of school’s name, search bar, reset button, logout button and table data of every student. The table consists of name, student\_id, ACT and SAT scores. Search bar can be used to search student according to name and score. Individual student details can be viewed separately.

# In Scope

Firstly we have to perform one level of functional testing for all the new features included in the sprint.

1. Login in system

* School Administrator should be able to login in system using valid credentials and can be able to logout from the system.

1. Viewing student list

* School Administrator should be able to view Student List with their Names, Student\_id, Scores(ACT,SAT), etc.

1. Viewing individual student scores

* School Administrator should be able to view scores of individual students.

1. Searching student name score in table

* School Administrator should be able to search student’s name, score.

1. Exporting student details in CSV format

* School Administrator should be able to export Student details in CSV format.

# 

# Out of scope

* Performance Testing

The system is not complex, so performance testing is not required in this sprint.

* Security Testing

The requirement in this sprint does not highlight the security of the system, so it might be done in any other sprint.

# Testing Strategy/Approach

A test strategy is an outline that describes the testing approach of the software development cycle.

**Functional Testing**

The system is tested against the functional requirements/specifications. Features are tested by providing them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is not concerned with how processing occurs, but rather, with the results of processing. It simulates actual system usage but does not make any system structure assumptions.

For example Login module, Main dashboard module, Individual student detail module will be tested using positive and negative testing whether the system is responding to the valid and invalid inputs.

**API testing**

API testing will be performed to fetch data from database by sending a GET request to check whether the system will respond or not. If the system responds to the request, status 200 ok verifies that data has been successfully fetched.

# Test Item

* Login
* Student list
* Student details
* Searching
* Sorting
* CSV export

# 

# Tool

POSTMAN (for API testing)

# Assumption / Risks

## ASSUMPTION

1. The test plan will be prepared to meet the requirement of the client.
2. The test plan will be prepared in such a way that it can validate user stories.
3. Login will be successful by providing valid credentials.
4. Searching and Sorting names, score will be covered in test plan.

## RISKS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Risk | Impact | Priorities | Mitigation Strategy |
| 1 | Change in requirement | High | High | Communicating with client and stakeholder about requirement change and make some changes in test plan. |
| 2 | Estimation properly | High | Medium | Estimation planning is done before conducting test cases. |
| 3 | Testing Environment | High | Medium | Testing system in different operating system to make sure that it can run smoothly. |
| 4 | Test data prepared may not be sufficient. | High | High | Communicating with stakeholders and validating if any data are missing with client. |

# Test Deliverables

* Test Plan document
* Test Results
* Test Cases
* Test Data
* Sanity Checklist
* Test Matrix
* Bug Report

# Testing Environment

Testing the system in different environment

|  |  |
| --- | --- |
| Environment |  |
| Browser | Safari, Chrome |
| Operating System | Linxus, Windows |

# Bug Report

It stores all the information needed to document, report and fix problems that occur in an application. It keeps track of all the bugs that are found during the execution of the test cases and mark them according to their priority.