

Test Plan Documentation

PRISM Behavioral Management

Prepared By
SMNEK for CS-472-1 Course

In the Year of Npm 9.6.0 release 2023 May 9.

Table Of Contents

1. Introduction
2. Objectives & Task
 - a. Objectives
 - b. Tasks
3. Team Members
4. Scope
5. Testing Strategy
 - a. Test Automation
 - b. Test Environment
 - c. Unit Testing
 - d. System and Integration Testing
 - e. User Acceptance Testing
 - f. Beta Testing
6. Hardware Requirements.
 - a. Desktop
 - b. Mobile
7. Environment Requirements
8. Testable Features
 - a. Features to be Tested
 - b. Features not to be Tested
9. Milestones / Deliverables
 - a. Test Schedule
 - b. Deliverables

1.0 Introduction

This software test plan outlines the comprehensive testing approach for PRISM Behavioral Management Software, an innovative and easy-to-use tool developed by SMNEK team for CS-472-1. This test plan aims to ensure the delivery of a high quality, user-friendly website through a thorough analysis of its functionality, performance, and data security and compliance. Through careful definition of test goals, scope, methodologies and environments. This plan helps layout a plan that guides the testing team through the entire development lifecycle, helping resolve potential bugs. Adherence to this test plan enhances the sites overall reliability, and client rating.

2.0 Objectives and Tasks

a. Objectives

PRISM is a web-based behavior support planning tool. This tool assists teachers in pre-secondary school in developing customised set of strategies for supporting the behavioral needs of any of their students. The test team is responsible for ensuring the product meets the clients specification, and its valid for the usecases with respect to its standard and validation.

b. Tasks

- i. Tested the backend data model unit.
- ii. Tested integrative units of end-points
- iii. Automated both frontend and backend actions.

3.0 Team Members

Name	Role
Sadie Grode	UI/UX Designer
Emma Foulk	Backend/Frontend Dev
Kamontat	Devops
Michael	Frontend/Backend Dev
Nana	QA Engineer/Automation

4.0 SCOPE

The scope of this testing covers the frontend and the backend components of the product whilst in development mode , before being shipped to production. The purpose of this test is to ensure functional correctness of the backend components as it relates to the datatypes of our data model objects represented in python django. Moreover, the frontend side of the test will include a manual end-to-end testing verify all specific functions and API calls are outputting the correct information. I would also checking the performance and rendering of the frontend with respect to API calls. Also, I will be checking the authentication feature.

5.0 TESTING STRATEGY

- a. Test Automation** : Selenium will be used to build automated test cases to verify the frontend elements of the project.

- b. **Test Environment** : Google chrome browser will be used as the initial test environment for the frontend. There must also be an adequate internet connection required to facilitate communication between the backend and frontend. Later other browsers will be used and hopefully testing would be scaled to different systems from mobile to tablet.
- c. **Unit Testing** : Unit testing is ensuring individual components of the app works as expected, and it is displaying the correct output.
- d. **System and Integration Testing** : Ensuring specific systems works together as expected. This is heavily utilized during API calls and rendering of the frontend, ensuring correct data is being displayed.
- e. **User Acceptance Testing** : Showcasing the progressed product to the user.
- f. **Beta Testing** : Final end-to-end testing before being pushed to production.

5.0 Hardware Requirements

- a. **Desktop** : The desktop should have a modern browser that supports the current javascript language to help render the information. Also, the desktop should have a strong connection to the internet as there are multiple API calls to render content.
- b. **Mobile** : For mobile a current smartphone that supports modern browsers with javascript enabled will suffice .

6.0 Environment Requirements

- a. A good IDE like vscode running a programming language that is supported by selenium. Since selenium would be used for most of the automation tests, it is essential that it uses a good language. For this automation, python would be used to enable the

selenium framework as it handles dynamic typing to facilitate faster development time.

8.0 Testable Features

a. Features to be tested.

- i. Authentication.
- ii. Form creation
- iii. Account creation
- iv. Form Deletion.
- v. Student Creation
- vi. Profile Update

b. Features not to be tested.

- i. Admin site of Django admin, since django is a tight knit system it would be hard to unravel under the hood to test the system.
- ii. Hovering on certain elements to display tooltips will not be tested.

9.0 Milestones/ Deliverables

a. Schedule

Task Name	Start	Finish	Effort	Comments
Test Planning	3/10/23	5/19/23	10	95% implemented
Review Requirements documents	3/20/23	3/27/23	Communicated with Michael regularly about requirements	

SetUp DjangoTest	3/17/23	3/20/23	Dataobjects passed the test cases	
SetUp Selenium	3/25/23	5/19/23	Learnt selenium to understand to use	
Produced Testcases for backend data objects using django test	3/25/23	4/7/23		Issue around duplication was rectified by Michael.
Automated Login page with Selenium	3/28/23	4/1/23		Run smoothly, had to update test as devs merged both username and email field.
Automated account creation	4/10/23	4/17/23	Implemented with selenium	Tested creating multiple accounts in one run to check the performance of the database
Automated form creation and student creation	4/12/23	4/30/23		Continuously had to revision the tests to meet up with dev changes.

b. Test Cases

Test cases are stored in the main repository in the **nana-production** branch for the frontend automation. Also for the backend, the the test cases

were merged with main, in the backend folder had to stop merging my branch with main as it caused some merged conflicts hence, the reason I had been merging on a different production.

c. Bug Report

Bug Issue	Description	Status
Duplication creation of multiple students with similar data	So in a teachers homepage, a student can be created multiple times with similar field	Resolved with Michael, and he stated that its not a big deal issue with respect to the backend.
Email Authentication not working in production	The system was not generating an authentication email for new users when prompted	Fixed
Survey form not branching	The form survey part of the project was not branching to the required sets of question.	Fixed
Wrong indexes associated with forms	Since form question elements were generated by backend logic they were tied to specific questions and some of the questions did not have the correct index, hence was generating wrong answers	Communicated with the dev team to rectify it.

