Dwebble:
Testing document and Specifications

Test Plan

Group 12

Elias Julian Marko Garcia

Ibrahim AbuAnz

Spring, 2020

Introduction

This test plan aims to provide information about the testing approach, testing criteria, test deliverables, and other relevant information on planning for the testing of the Alternate Schedule Application developed for a University. According to the system requirements specifications, the Alternate Scheduling application will help admin staffs of a university to generate multiple schedules for classes based on various rules and constraints on the availability of rooms, preferences of instructors, and relations with other courses in progress at any point of time. The application helps in reducing the manual effort and chances of errors in preparing schedules for university classes. The system also allows instructors to check for their personalized schedules time to time and setting preferences for upcoming classes.

The primary objective of the testing process is to verify and validate whether the proposed Alternate Scheduling application is working according to the system requirements specifications or not. The testing process will be focused on checking whether the system can generate multiple schedules by following the rules and preferences in real-time or not, whether an admin staff can add information about new classes, and rules in the system or not, and whether instructors can check schedules and set preferences or not. It will also check whether there is a secure sign-in process for all user through the front-end interface or not.

Terminology

Following terminologies will be used throughout this test plan.

Terminology	Meaning
Admin Staff	A university staff responsible to make schedules for classes and having administration-level access to the Alternate Scheduling application.
Instructor	Teachers who take classes at the university.
System	Refers to the Alternate Scheduling application being developed for the university.
Username and password	Unique and system-generated pair of information that enables a legitimate user to log in to the system. The username is the staff ID number registered with the university for all admin staff and instructors. And passwords are generated randomly.
User interface	The front-end web-based interface accessible to mobiles and desktop.
User	Either admin staff or an instructor based on their roles.

Items Tested

The list of items tested in this testing process has been given below in the table. The list is not exhaustive.

Test Cases	Description
1.1	The ability of a user to login to the system using their unique login credentials.
1.2	The ability of admin staff to add a new class in the system by specifying detailed class information
1.3	The ability of admin staff to add rules to follow while processing a schedule creation request.
1.4	The ability of admin staff to select from a list of possible schedules as generated by the system
1.5	The ability of an instructor to set preferences for an upcoming class
1.6	The ability of an instructor to check the personalized schedule at any point in time.

Items Not Tested

Some of the aspects of the system are beyond the scope of the testing process as planned in this document. Those are listed below.

- Checking whether the instructors are getting notifications automatically when a new class is added to the system and an instructor is selected for the class.
- Whether the user interface follows the usability and user experience design requirements or not.
- Whether the system is crashing or not
- Measurement of runtime metrics like time to create a schedule, and other performance metrics.

Approach

A manual approach is taken in the overall system testing process for the Alternative Schedule application development project. First of all, while writing code, automated syntax checkers and debuggers have been used to minimize the syntactical errors in coding. However, the manual system testing is required not only to check whether the logical and semantical aspects of the system is working properly or not.

To keep the testing time-limited and not extended, the testing has been done categorically. For example, while checking for the login function for different users, different login details will be used separately and tested whether the function is working properly or not. And while checking whether an admin staff can add a class successfully, it will also check whether the system is triggering a notification to the instructor assigned to the class or not.

Even though the testing is a part of each iteration, but that is mostly unit testing for each module developed in an iteration. There will be five such iterations. And the comprehensive manual system testing will be performed after all five iterations are finished. And in the comprehensive system testing, all functions that have been implemented will be tested.

Items Pass/Fail Criteria

All functions according to the system requirement specification document will be tested and the outcomes of the testing will be marked along the tests done. If any bug or error is found during testing, then the details will be shared with the development team. Once the development team fixes the issue, then it will be tested again to check whether it is performing as intended or not.

Test Deliverables

There are three deliverables, a document on a test plan, a document on test specifications, and test reports. The documents will give a consolidated overview and details on the test done, the outcome of the tests, and information about bugs, changes, and incidents.

Testing Tasks

The details of the deliverables developed while performing the testing tasks have been summarized below,

Deliverables	**Activities**
Test Plan	• Outlining the planning purpose, aim, approach and defining the deliverables and an overall approach for testing. • Deciding the testing methods to follow • Deciding the testing approach follow

	• Deciding on what items to test and what items not to test
Test Specifications	• System requirements analysis for verification and validation • Preparing test cases as outlined and planned in the test plan document
Test Reports	• Execution of the test plan according to test specifications • Documenting about incidents, bugs and errors • Change request sent to developers • Coordinating with a developer to fix the bugs and errors • Documentation of the change and outcomes