

# Test Plan

---

**Document owner:** Mateo Picari is responsible for developing and maintaining this document.

**Date:** April 26, 2020

**Version:** 1.0

## Revision History

Version	Date	Name	Description
1.0	26/04/2020	Mateo Picari	Initial document

## Identifier

MarconiTT Software Test Plan

## 1 Introduction

MarconiTT is a web application available to desktop and mobile users and accessible through a browser. It can be reached both from inside and outside the school network. The application allows for authorized teachers, students and staff members to book school classrooms and labs. This document outlines the testing procedures for MarconiTT.

### 1.1 Items and Features Tested

The features tested include the containers management, the timetable update and the user website.

### 1.2 Test Specifications

This section provides the test cases to be carried out for the MarconiTT Application. It includes system tests performed by a team of testers to verify that the system performs as expected and eventually report issues. Each item to be tested is represented by an individual test case. The complete testing procedures are outlined in the test specifications document.

### 1.3 Software Project Management Plan

The software project management plan reconciles the time schedule, estimated effort, number of iteration, the release of candidates with the system testing and the product acceptance plan.

## 2 Items and features tested

### 2.1 Containers Management

Containers Management testing takes place in the system testing phase. The team will test the system by starting containers and running the application, by stopping containers, cleaning the

cache and temporary data and restarting the application. Test items:

- Docker startup
- Clean restart

## 2.2 Timetable Update

Timetable Update testing takes place in the system testing. The team will test the automatic process of detecting the timetable files and updating the whole timetable on the database.

Test items:

- Timetable update

## 2.3 User Website

User Website testing takes place in the integration and system testing. The team will test functions of the project during the integration of additional features, and then will test them again after the new features are added, in order to make sure that everything still functions correctly.

System testing will test all the aspects of the system, particularly the new features that have been added.

Test items:

- Multiple day booking

# 3 Approach

## 3.1 Strategy

The strategy is for the team members to act as users of the application. They will carry out activities similar to the ones carried out by the regular users of the application. These activities include viewing the timetable to see which classroom is already booked or not, logging in as an allowed user and booking classrooms or labs in different ways, inserting and deleting events, freeing up resources. Team members will also create unexpected scenarios such as invalid login, invalid bookings or invalid insertions of events to test how the system will respond to such scenarios.

## 3.2 Techniques/Tools

Testers will perform testings of the whole system both from the user side and the administration side, which means that they will test the final application and they will also go through the code to find bugs and errors.

## 3.3 Test Goals

The goals of these testings are to ensure that everything works perfectly, especially the added features. Testers will also try to find bugs and discover errors that may cause system failures, that will eventually need to be corrected in order to deliver an application without errors.

## 4 Test Deliverables

- A test plan that delineates the testing phase of the development process
- A test specification document detailing steps in the testing procedures

## 5 Testing Tasks

Testing tasks for integration and system testing are outlined in the test specification document.

## 6 Schedule

At the conclusion of each iteration, the product needs to be tested in an environment similar to the one used on the Production server. In the fourth iteration, where the software on edu-x08 is updated at its latest version, production tests are required in order to make sure that the software is ready to be finally released. Test cases are outlined in the test specification document for the iterations.

## 7 Required Resources

Human resources:

Farkas Raul

Moretto Pietro

Picari Mateo

Pozzani Samuele

Material: Python, HTML, CSS, JSON, YAML, Angular JS, Express(Node), MySql, Docker and Docker-Compose for development, Visual Studio Code as the development tool, Linux as the operating system, GIT repository for the version control, Markdown(.md) for documentation.