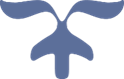


TEAM ZAFE



**MusalaSoft Sprint project**

**School Form**

Test Plan

***Shape

Description generated with very high confidence***

Contents

[1. Introduction 3](#_Toc65003804)

[2. Entry and exit criteria 3](#_Toc65003805)

[2.1 Entry criteria 3](#_Toc65003806)

[2.2 Exit critertia 3](#_Toc65003807)

[3. objectives and Tasks 3](#_Toc65003808)

[3.1 Objectives 3](#_Toc65003809)

[3.2 Tasks 3](#_Toc65003810)

[4. Scope 4](#_Toc65003811)

[4.1 Manual testing 4](#_Toc65003812)

[4.2 Automated testing 4](#_Toc65003813)

[5. Approach 4](#_Toc65003814)

[5.1 Used texting techniques 4](#_Toc65003815)

[5.2 Automation approach 4](#_Toc65003816)

[6. Testing process 4](#_Toc65003817)

[6.1 Test deliverables 4](#_Toc65003818)

[6.2 resources 4](#_Toc65003819)

# Introduction

* This documentation provides a plan for testing a School form, from when you choose your role in a school. You can be a student, teacher or admin. It is suitable for people of all ages and does not require any specific skills to begin with.

# Entry and exit criteria

## 2.1 Entry criteria

* Access to the application is provided.
* Code development has been paused until QA testing is finished.

## 2.2 Exit critertia

* Every function is tested successfully.
* There are not any critical issues.

# objectives and Tasks

## 3.1 Objectives

* The objectives of this testing process are to provide verification that at least 95% of the functionalities of the application are suitable for use and do not have any major issues or bugs.

## 3.2 Tasks

* The foremost tasks that must be completed to successfully test the program and assure its bug free status will be:
  + Defining the most important elements to be tested
  + Designing the various test cases
  + Performing manual and automated testing
  + Preparing test and bug reports

# Scope

## 4.1 Manual testing

* Game interface
* Menu options
* Data input
* Verify that user data can be entered
* Verify that different error messages are displayed for different occasions
* Verify that correct values by requirements are considered acceptable by the code

## 4.2 Automated testing

* Basic functions for user input
* Game function algorithms

# Approach

## 5.1 Used texting techniques

* Functional testing
* Unit texting
* Usability texting

## 5.2 Automation approach

* Unit testing for C++ with Native Unit Test Projects from Visual Studio

# Testing process

## 6.1 Test deliverables

* Excel file with description of the test suits filled with different test cases
* Test plan

## 6.2 resources

* Microsoft Unit Testing Framework for C++ for testing the functions
* Excel as test case management tool
* GitHub for reporting problems