MASTER TEST PLAN

Contents

[1. Introduction 2](#_Toc66726122)

[2. Objectives and tasks 2](#_Toc66726123)

[Objectives 2](#_Toc66726124)

[Tasks 2](#_Toc66726125)

[3. Scope 3](#_Toc66726126)

[Features that will be tested: 3](#_Toc66726127)

[1. Manual testing 3](#_Toc66726128)

[2. Exploratory testing 3](#_Toc66726129)

[Features that won’t be tested: 3](#_Toc66726130)

[4. Approach 3](#_Toc66726131)

[5. Testing process 4](#_Toc66726132)

[Test Deliverables 4](#_Toc66726133)

[Responsibilities 4](#_Toc66726134)

[Resources 4](#_Toc66726135)

[6. Entry and exit criteria 4](#_Toc66726136)

[Entry Criteria 4](#_Toc66726137)

[Exit Criteria 5](#_Toc66726138)

[7. Estimation 5](#_Toc66726139)

[8. Environment requirements 5](#_Toc66726140)

# Introduction

This document gives a detailed plan of the project testing. The project is a C++ based application that creates and manages different teams in a school. It consists of the objectives and tasks of the testing process, the testing scope, approach and hardware requirements.

# Objectives and tasks

## Objectives

The test objectives are to verify the functionalities of the tool to guarantee its usability.

## Tasks

The main tasks that are going to be completed in accordance with the test objectives will be:

1. Performing a detailed analysis of the application functionalities
2. Setting the appropriate testing levels and types
3. Executing manual tests
4. Writing a Test plan

# Scope

## Features that will be tested:

### Manual testing

* The functionalities of the program

### Exploratory testing

* The input of the program
* The output of the program

## Features that won’t be tested:

1. The interface of the application

# Approach

This section describes the overall approach to testing. For each major group of features or feature combinations, it specify the approach which will ensure that these feature groups are adequately tested, as well as the tools and techniques used in the testing.

1. Test types and design techniques

* Functional testing
* Usability testing
* Case testing
* Exploratory testing

1. Tools

* C++
* Visual Studio 2019 debugger

# Testing process

## 

## Test Deliverables

1. Summary Test Report
2. Issue report

## Responsibilities

1. Design high-level test cases
2. Test Case Manual Execution
3. Bug and test case summary reports
4. Test case execution

## Resources

1. Visual Studio as source code management tool
2. Visual Studio as a tool for bug logging/tracking
3. Word and Excel for the Test plan and Test cases

# Entry and exit criteria

## Entry Criteria

1. Software requirements are provided
2. Functionality is deployed on environment
3. Required access is provided
4. Test procedures defined
5. Completion of Unit testing

## 

## Exit Criteria

1. All tests are covered
2. All test cases are executed and pass
3. The program is functioning properly

# Estimation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Type | Complexity | Order | Days |
| Test plan | Independent | 3 | 1 | 2 |
| Exploratory testing | Dependent | 3 | During the whole process | - |
| Manual testing | Dependent | 5 | 2 | 2 |

# Environment requirements

OS: Windows 7, 8.1, 10; 64-bit

RAM: 2GB minimum; 8GB recommended

CPU: 1.8 GHz or faster