MASTER TEST PLAN

## TABLE OF CONTENT

Contents

[TABLE OF CONTENT 1](#_Toc66458196)

[INTRODUCTION 1](#_Toc66458197)

[DEFINITIONS/ACRONYMS 2](#_Toc66458198)

[ENTRY AND EXIT CRITERIA 3](#_Toc66458199)

[Entry Criteria 3](#_Toc66458200)

[Exit Criteria 3](#_Toc66458201)

[OBJECTIVES AND TASKS 3](#_Toc66458202)

[Objectives 3](#_Toc66458203)

[ROLES AND RESPONSIBILITIES 4](#_Toc66458204)

[SCOPE 4](#_Toc66458205)

[APPROACH 5](#_Toc66458206)

[Methodology 5](#_Toc66458207)

[TESTING PROCESS STRATEGY 5](#_Toc66458208)

[Test Deliverables 5](#_Toc66458209)

[Responsibilities 5](#_Toc66458210)

[Resources 6](#_Toc66458211)

[Estimation 6](#_Toc66458212)

[QA role in the test process 6](#_Toc66458213)

[Testing Types 6](#_Toc66458214)

[Severity List 7](#_Toc66458215)

[Testing tools list 7](#_Toc66458216)

[ENVIRONMENT REQUIREMENTS 7](#_Toc66458217)

## INTRODUCTION

This document describes the testing process of the console application of TeamsOnBudget. The application is intended for teachers or school administrators and allows them to:

* Easily manage the school work
* To filter and find students in their school more easily
* To assign them to different teams
* Assign projects to those teams

## DEFINITIONS/ACRONYMS

1. APP – application
2. TCL – Test Case Lab
3. CRUD – Create, Read, Update, Delete

## ENTRY AND EXIT CRITERIA

### Entry Criteria

1. Software requirements are provided
2. Clear idea of the whole application is defined and approved
3. Required access is provided
4. Completion of Unit testing
5. Code freezes and there are no further changes

### Exit Criteria

1. All happy paths are covered
2. All test cases are marked as completed and invalid data cannot be entered
3. All CRUD operations are working as expected
4. All GitHub issues are closed
5. Most of the functions are tested
6. Application is validated and all bugs and defects are fixed
7. The schedule has been achieved
8. The deadline is reached
9. All TCL cases pass successfully
10. System testing has been performed

## OBJECTIVES AND TASKS

### Objectives

Primary objectives

The main test objectives are to ensure that all application functionalities of the TeamsOnBudget application and to guarantee them after the deployment process. The test plan is used to define the entry criteria, exit criteria, time taken for the implementation, different responsibilities of the different roles, and bug reporting. At the end of the development part, users should find out that the application has met all required criterias and covered their expectations, which are soft and understandable design, successful CRUD operations through navigated menu and so on.

Secondary objectives

The secondary objectives are to identify and expose all possible bugs, risks, validation and range errors, exceptions and issues. The application must be edge tested in every possible situation. To be completed as a main secondary objective, careful and methodical testing is required to ensure that functionalities are working properly, including CRUD operations, integrated user experience and more.

Tasks

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

1. Performing manual tests (mostly for the data input and user experience);
2. Performing automatic native unit tests on most of the functions (if possible);
3. Creating test cases;
4. Application is working on other operating systems such as GNU/Linux and its distributions

# ROLES AND RESPONSIBILITIES

|  |  |
| --- | --- |
| Role | Responsibilities |
| SCRUM Master | Acts as a primary contact between the team members. Uses agile methods to schedule meetings and to ensure success of the team and the project itself. |
| Back-end Developer | Responsible for the main logic, realization of the app, CRUD operations and more. Sometimes he might be required to fix issues. |
| Front-end Developer | Responsible for the design of the console application. Must care for the user experience part of the application |
| QA Engineer | Responsible for writing and running automatic and manual test cases and to ensure that the app is functioning and working properly. May help the Back-end developer |

# SCOPE

1. Manual testing
   * User experience
   * Application starts
   * Menu options are working correctly
   * Correct code typing
   * Correct info messages
   * Correct and not offensive error messages
2. Automatic testing

* Updating data functions
* Validation functions
* Reading functions
* Functions that process other kind of data

## APPROACH

### Methodology

White Box testing methodology will be used as well as the triple A (Arrange, Act, Assert) in order to test and verify full application functionalities. All test cases will be created during testing and recorded after that. The tester will know what the developer has written as a code, to some extent. Testing approach is as follows: An issue or bug is found by the QA, after being tested. Then the QA Engineer tries to fix it or creates an issue and reports to the developer of the unit where the issue was found. There will be weekly discussions about all current bugs.

## TESTING PROCESS STRATEGY

### Test Deliverables

1. Native unit testing framework with C++
2. Excel Summary Tested cases
3. Word Test Plan report
4. Test Case Lab

### Responsibilities

1. Development Team
   1. Developing the required task
   2. Fixing bugs if they find such
2. QA Role
   1. Finding test cases
   2. Test case manual execution
   3. Test case automation execution (With Native unit testing framework)
   4. Bug and test case excel summary reports

### Resources

1. Excel as a test case management and writing tool
2. GitHub as source code management tool
3. GitHub as an issue and bug tracker
4. Native unit testing as a main testing framework

# Estimation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Type | Complexity | Order | Days |
| Test plan | Independent | 3 | 1 | 2 |
| High-level test cases | Dependent | 3 | 2 | 4 |
| UI Tests | dependent | 1 | 4 | 2 |
| Test Report | Independent | 4 | 5 | 2 |
| Bug Report | Independent | 3 | 6 | 2 |
| Script | Independent | 2 | 7 | 1 |
| Manual Testing | Dependent | 4 | During the whole process of the realisation |  |
| Presentation | Independent | 5 | 8 | 1 |

Time required to test the product: 18

### QA role in the test process

1. Preparing Test Cases: QA will prepare a test case during some test periods.
2. Creating Test Data:

Test data will be created and used to test the application based on different scenarios.

Test results will be put in the test suite document.

QA will log all found bugs and issues.

QA will report all found issues to the developer

1. Deployment

When all bugs and issues are fixed and the test cases are covered, QA will report to the team that the app is ready for to be deployed.

### Testing Types

1. White Box Testing: QA will previously know how the code and architecture works. QA will focus on the functional requirements of the app and will know what to test and how.
2. GUI Testing (Menus testing): This testing will include the testing of the GUI or Frontend part of the application so to say. It will cover spelling mistakes, soft error messages, user comfortability and experience, etc
3. Performance Testing: Check the optimal time that it takes to execute different CRUD operations
4. Functional Testing: This will test whether the application meets the task criteria.It will be also used in order to find out unexpected and unwanted behavior. Provides correctness, accuracy and reliability.

### Severity List

|  |  |  |
| --- | --- | --- |
| Severity ID | Severity | Severity Description |
| 1 | High | The application’s crashes and requires restart. Bugs cause serious problems and fails functionality |
| 2 | Medium | Incorrect functionality. Simple work may fix the problem |
| 3 | Low | Simple issues which can be fixed in less time |

### Testing tools list

|  |  |
| --- | --- |
| Process | Tool |
| Test Case Report | Excel, Test Case Lab (TCL) |
| Unit tests | Native unit testing with C++ |
| Test Report | Word 2016 |

## ENVIRONMENT REQUIREMENTS

1. Requirements:

For Windows - Windows 7 or upper versions and up-to date Visual Studio

For Linux – Up-to date g++ compiler, works on most distributions