Foster Grandparent Program

Grandparent Education Management System

Quality Assurance Procedures

Version 1.0

March 23, 2023

Prepared by Quality Assurance Team

Table of Contents

[1.0 Introduction](#_Toc130305953)

[1.1 Purpose](#_Toc130305954)

[1.2 Scope](#_Toc130305955)

[1.3 Definitions and Acronyms](#_Toc130305956)

[2.0 Procedures](#_Toc130305957)

[2.1 Bug Report](#_Toc130305958)

[2.1.1 Purpose](#_Toc130305959)

[2.1.2 Documenting a Bug](#_Toc130305960)

[2.1.3 Reporting a Bug](#_Toc130305961)

[2.2 QA Request](#_Toc130305962)

[2.2.1 Purpose](#_Toc130305963)

[2.2.2 Submitting a QA Request](#_Toc130305964)

[2.3 Post Merge Request](#_Toc130305965)

[2.3.1 Purpose](#_Toc130305966)

[2.3.2 Documenting a Bug](#_Toc130305967)

[2.3.3 Reporting a Bug](#_Toc130305968)

[3.0 Testing](#_Toc130305969)

[3.1 Testing Process](#_Toc130305970)

[3.1.1 User Interface](#_Toc130305971)

[3.1.2 User Interface XAML](#_Toc130305972)

[3.1.3 Backend 7](#_Toc130305973)

[3.1.4 Database 7](#_Toc130305974)

[3.2 Tools and Technologies 7](#_Toc130305975)

[3.2.1 Github 7](#_Toc130305976)

[3.2.2 Visual Studio 2022 7](#_Toc130305977)

[3.2.3 SQL Express](#_Toc130305978)

[3.2.4 MSTest](#_Toc130305979)

[3.3 Best Practices and Guidelines](#_Toc130305980)

[3.3.1 Comments](#_Toc130305981)

[3.3.2 Variable Naming](#_Toc130305982)

# Introduction

* 1. Purpose

The purpose of this document is to provide a comprehensive description of the processes and procedures employed by the Quality Assurance team throughout the development cycle of the Grandparent Education Management System. This document outlines the testing procedures for every aspect of the application, including but not limited to the User Interface, Backend, and Database. By following these procedures, the QA team ensures that the final product is thoroughly tested and meets the required standards of quality and functionality. Additionally, this document provides definitions and acronyms to ensure clarity and understanding throughout the development process.

* 1. Scope

The scope of this document encompasses the Quality Assurance procedures and processes used by the development firm's QA team for the Grandparent Education Management System. This document outlines the testing procedures for each aspect of the application, including but not limited to the User Interface, Backend, and Database. It does not include development processes or procedures followed by other teams or departments within the organization. This document is intended to serve as a guide for the QA team and any other relevant parties involved in the development of the Grandparent Education Management System.

* 1. Definitions and Acronyms

|  |  |
| --- | --- |
| Quality Assurance – (QA) | The process of ensuring that a product or service meets the desired level of quality by identifying and addressing defects, bugs, or other issues before the final release. |
| User Interface – (UI) | The visual and interactive components of a software application or website that enable users to interact with the system and accomplish their tasks. |
| Application Programming Interface – (API) | A set of programming instructions and standards for accessing a web-based software application or web tool. An API allows different software applications to communicate with each other. |
| Database – (DB) | A collection of data that is organized in a way that allows efficient retrieval and manipulation of the data. Databases are used to store and manage information in a software application. |
| Software Development Life Cycle – (SDLC) | The process of designing, developing, testing, deploying, and maintaining a software product or application. |
| Continuous Integration/Continuous Deployment – (CI/CD) | A set of practices that automate the process of building, testing, and deploying software. Continuous Integration ensures that changes are tested and integrated into the codebase as quickly as possible, while Continuous Deployment automates the release of the software to production. |
| Integrated Development Environment – (IDE) | A software application that provides comprehensive facilities for software development, including a code editor, compiler or interpreter, and debugging tools. |
| Version Control System – (VCS) | A software tool used to manage changes to a codebase over time. VCSs track every modification made to the codebase, enabling developers to review and roll back changes as needed. |
| Test-Driven Development – (TDD) | A software development approach that involves writing tests before writing code, with the goal of producing more reliable and maintainable software. |
| Structured Query Language – (SQL) | A standard language used for managing relational databases and performing various operations on them, such as adding, updating, deleting, and retrieving data. |
| GitHub | GitHub is a web-based platform for version control and collaborative software development. It allows users to host and review code, manage projects, and build software with others. |
| MSTest | A unit testing framework commonly used for testing .NET applications to ensure code quality is up to standards and help deliver a more reliable product. |

# Procedures

* 1. Bug Report
     1. Purpose

This section describes the process for reporting bugs or defects in the Grandparent Education Management System. It outlines the steps involved in submitting a bug report, including how to identify the issue, document it, and communicate it to the appropriate stakeholders.

* + 1. Documenting a Bug

After a bug has been discovered, the person who identified it should document the following information: the location where the bug occurred, a description of the bug, the steps to reproduce the issue, the expected result, the actual result, and the file or module where the bug occurred.

* + 1. Reporting a Bug

To report a bug, use the online "Bug Report" form located on the "Issues" page of the GitHub repository. The submitter must provide a brief title that describes the bug, along with a detailed description of what happened, the steps to reproduce the issue, the expected and actual results, any relevant logs, and any other information related to the bug.

Graphical user interface, text, application

Description automatically generated

## QA Request

* + 1. Purpose

This section covers the procedures for requesting quality assurance testing for a specific feature or component of the Grandparent Education Management System. It includes the criteria for submitting a QA request, the steps involved in the QA process, and the expected turnaround time for completing a QA review.

* + 1. Submitting a QA Request

You can request a QA review of any part of the application at any time. To make a request, use the online "QA Request" form located on the "Issues" page of the GitHub repository. The submitter should provide a brief title that describes the purpose of the request, a description of the changes made, any specific testing scope, relevant environmental properties, a deadline, and any other pertinent information.

## Post Merge Request

* + 1. Purpose

This section outlines the procedures for requesting post-pull request reviews for changes made to the Grandparent Education Management System. It covers the criteria for submitting a post merge request, the steps involved in the review process, and the expected turnaround time for completing a post merge review.

* + 1. Documenting a Bug

After a bug has been discovered, the person who identified it should document the following information: the location where the bug occurred, a description of the bug, the steps to reproduce the issue, the expected result, the actual result, and the file or module where the bug occurred.

* + 1. Reporting a Bug

To report a bug, use the online "Bug Report" form located on the "Issues" page of the GitHub repository. The submitter must provide a brief title that describes the bug, along with a detailed description of what happened, the steps to reproduce the issue, the expected and actual results, any relevant logs, and any other information related to the bug.

# Testing

## Testing Process

* + 1. User Interface

When it comes to testing a user interface, there are two key aspects to consider: functionality and similarity to the proposed user interface storyboard. In terms of functionality, it's important to ensure that the interface performs all the necessary tasks and functions as intended. Meanwhile, similarity to the proposed storyboard involves ensuring that the final product matches the initial design as closely as possible. This can involve comparing the final interface to the storyboard and making any necessary adjustments. Overall, thorough testing of both functionality and similarity to the storyboard is crucial to ensure a high-quality user interface that meets user needs and expectations.

* + 1. User Interface XAML

Testing the XAML files in a C# program is an essential part of ensuring the functionality and user interface of the program. XAML (eXtensible Application Markup Language) is a language used to define the user interface of Windows Presentation Foundation (WPF) applications in C#. Testing XAML files involves validating the markup syntax, verifying the layout, and ensuring that the controls and data bindings are working as intended. Testing can be done manually by running the application and interacting with the user interface or by reviewing the XAML code itself. The purpose of testing the XAML files is to identify any defects that may affect the user experience, such as inconsistent or missing elements, incorrect alignments or sizes, or incorrect behavior of user input. Ensuring that the XAML files are well tested and functional helps to create a positive user experience, increases the usability of the application, and reduces the likelihood of errors or bugs.

* + 1. Backend

Testing the backend functionality of a system is an essential step in ensuring that the system performs as intended. This type of testing involves examining the functionality of the components of the system that are not directly visible to the end-user. During backend testing, a tester would typically test how well the system handles inputs, processing, and outputs. Additionally, the tester would check for any bugs, errors, or performance issues that might arise in the backend. The goal of this type of testing is to ensure that the system can function as expected and provide reliable and accurate data to the user interface. Overall, testing the backend functionality is a crucial aspect of ensuring the overall quality and reliability of the system.

* + 1. Database

Testing a database of a system is a crucial step in ensuring its accuracy, efficiency, and security. This type of testing involves checking the database's data integrity, performance, and scalability. The tester needs to validate that the data stored in the database is accurate, consistent, and complete, and that it can be accessed and modified as required. Furthermore, the tester needs to verify that the database can handle the anticipated amount of data and user requests, and that it performs efficiently and without errors. Overall, testing the database is essential for ensuring the reliability, accuracy, and security of the system's data, which is vital for the system's successful operation.

## Tools and Technologies

### GitHub

GitHub is a tool for version control in software development projects. It allows developers to track changes to their code over time, collaborate with others, and manage different versions of their software. To use GitHub for version control in a project, developers first create a repository to store their code. They can then make changes to the code, commit those changes to the repository, and push the changes to the remote repository on GitHub. This creates a history of changes that can be reviewed, rolled back, or merged with other branches. GitHub also offers tools for managing issues and pull requests, enabling developers to collaborate and review each other's work, and follow QA procedures.

### Visual Studio 2022

Visual Studio 2022 is an integrated development environment (IDE) that is widely used by developers for .NET application development. This tool provides a wide range of features that enable developers to build, debug, and deploy applications easily and efficiently. Visual Studio 2022 offers the ability to work with various programming languages, frameworks, and technologies, including the ones we use in this project, .NET, C#, and Entity Framework. It also includes numerous tools and services that simplify the development process, such as the live debugger, Git integration, and Azure cloud integration, all of which we deploy in our development environment.

### SQL Express

SQL Express is a free, lightweight, and easy-to-use version of Microsoft's SQL Server database management system. It is designed specifically for developers and provides a great platform for building and testing database applications. SQL Express offers a range of features that allow developers to create, edit, and manage databases, tables, and queries. It supports a wide range of data types and includes many of the same features as the full version of SQL Server, such as stored procedures, triggers, and views. SQL Express also integrates seamlessly with Visual Studio 2022, making it a popular choice for .NET developers, and perfect for our use case and project.

* + 1. MSTest

MSTest is a unit testing framework that is commonly used for testing .NET applications, including desktop applications. It allows the QA team to write automated tests to verify the functionality of the application, ensuring that it meets the expected requirements and that any changes made to the code do not introduce new bugs. MSTest also provides features such as test categories, data-driven testing, and test initialization and cleanup methods, which help to streamline the testing process and make it more efficient. By using MSTest, the QA team can save time and resources, improve the overall quality of the application, and ultimately deliver a more reliable product to the end-users.

A screenshot of a computer

Description automatically generated with medium confidence

## Best Practices and Guidelines

* + 1. Comments

The commenting style chosen by the development team is a combination of block comments as headers, for the whole file. As well as triple slash XML for additional information. The block header should contain the name of the developer, the purpose of the file, and dates. Function header comments will be in XML style and summarize the purpose of the function and parameters used. These should be homogenous throughout the project.

* + 1. Variable Naming

The development team has decided on using the following standards: Pascal case for class names, Hungarian notation for primitives, and camel case for comments. The notation standard should be consistent throughout the entire project.