Software Quality Assurance Plan

Office Hours Application

**Prepared by:**

Andrew McCracken and James Gruss, Jr.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Issue** | **Description** | **Author** |
| 10/19/2016 | 1,0 | Initial Version | JG, AM |
| 11/30/2016 | 1.1 | Updates: Sec 2 | JG |
| 12/16/2016 | 1.2 | After Testing Updates | JG, AM |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. Purpose

2. Reference Documents

3. Management

4. Documentation

5. Standards, Practices, Conventions, and Metrics

6. Software Reviews

7. Test

8. Problem Reporting and Corrective Action

9. Tools, Techniques, and Methodologies

10. Media Control

11. Supplier Control

12. Records Collection, Maintenance, and Retention

13. Training \_

14. Risk Management

# **1. Purpose**

The purpose of this Software Quality Assurance Plan (SQAP) is to define and articulate the methodologies, procedures, and activities by which the maintenance and extension of the Office Hours Sytem will be conducted in order to ensure the quality of the end product such that it, first, best conforms to its previously defined requirements and, second, best satisfies the needs of the end user. This includes the database and related interface from which the Office Hours System retrieves its information on the office hours of professors. The Office Hours system was initially created by a team of four (4) student requirements engineers and four (4) student software engineers.

# **2. Reference Documents**

This section provides a complete list of documents referenced elsewhere in the text of the

SQA plan.

C# Coding Conventions: https://msdn.microsoft.com/en-us/library/ff926074.aspx

# **3. Management**

The organizational structure of the project is small in relation to other software projects. The two members of the software quality assurance team are:

* Andrew McCracken
  + Specialty: user interface design,
* James Gruss
  + Speciality: user-driven development, architecture, interface definition, source control, inspection.

Each team member is equally responsible for the successful fulfillment of the purpose of this document. Each member will be expected to perform a reasonably equal amount of tasks.

The main stakeholders of the project are:

Kurt Spence: Gannon Applications Manager

Student Representatives

# **4. Documentation**

The full project is available at: [https://github.com/jgrussjr/cis315\_OfficeHour](https://github.com/jgrussjr/cis315_OfficeHours)s.

A live, functional site exists at: http://206.180.208.124/welcome.aspx

A streamlined version of the codebase, documentation, requirements, and other materials is included with this document.

One can find the following in the project folder. Paths are prefixed by: “OfficeHoursSystem\”.

* Requirements Specification: “requirements\VisionScopeBaseline\_V2.1.docx”
* Product Backlog: “ProductBacklog.docx”
* Initial Test Cases: “test\_cases\\*”

Each document will be reviewed by the Software Quality Team in order to ensure that the purpose of this document is fulfilled.

# **5. Standards, Practices, Conventions, and Metrics**

**Conventions**

In creation of new code, the *C# Coding Conventions* (see referenced documents) given my Microsoft will be used to provide consistency and best practice coding. Any refactoring of code, newly written or old, should also conform to these conventions.

**Metrics**

Completion Rates: Recorded as binary metric (1=Task Success and 0= Task failure). If users cannot accomplish their goal, nothing else really matters. This will measure usability.

* Find a professor’s office hours.
* Send an office hours request.
* Professor responds to request
* Student receives acceptance or denial if professors sends a reply
* Goal: 95% Completion Rate

Task Time: Record how long it takes a user to complete a task in seconds and/or minutes. Start task times when users finish reading task scenarios and end the time when users have finished all actions (including reviewing). An incomplete task will not be included in this metric. This is a usability metric

* Find a professor’s office hours. Goal: 1 minute
* Send a office hours request. Goal: 2 minutes

Bugs per line of code per component found in testing: Recorded as a decimal. Record the number of lines of code in a logical software component and record the number of errors discovered that result in failures of test cases. The goal is a maximum goal of .04.

# **6. Software Reviews**

Requirement Review: A formal requirement review will be conducted with the stakeholder, Kurt Spence, at the completion of the project.

Code Review: Before new functionality is added to the live project, a code inspection peer review session will be conducted. These code reviews will be done by the developers, who each act as all three recommended professionals for code reviews: designers, coders/implementers, and testers. The mentioned C# coding conventions will be used.

# **7. Test**

The existing code has been tested using manual tests for the front end and mostly automated tests for the backend. However, more testing will need to be done as there are existing bugs. Any new functionality added to the system will be thoroughly tested using similar techniques. Previous test cases and any new test cases can be found in the “test\_cases” folder of the Office Hours System project folder.

* Functionality to be Unit Tested:
  + Login Manager (LoginManager.cs)
  + Create New User
  + Check Password
  + Email Sender
* Functionality tested (additionally) via System Acceptance testing:
  + Welcome Page
  + Registration Page
  + Calendar Page
  + Unit test components in system context

# **8. Problem Reporting and Corrective Action**

The product backlog will host all currently known problems and their status. As problems are resolved, they will be updated in the backlog. Any new code, diagrams, or other models and specifications that have been approved through testing will be pushed to the project on GitHub.

# **9. Tools, Techniques, and Methodologies**

GitHub - Used for source code management and version control. GitHub supports SQA by hosting the current “deployable” project, which will have been thoroughly tested before being pushed.

Usability tests with users will be conducted to verify the system, as well as find bugs and weaknesses in the user interface.

# **10. Media Control**

The system will be tested and deployed on a Gannon ITS run server, which is physically under their control. The ITS staff will ensure the physical security of the server, as well as updates to the server.

# **11. Supplier Control**

Previously created code will be re-validated against the existing test cases within the existing project in order to ensure the system still fulfills its goals. New test cases, as well, will be created. Likewise, the reviews, as outlined in the reviews section, as well as our metrics, will be used to validate existing software pieces.

# **12. Records Collection, Maintenance, and Retention**

Github will be used to record and retain the software and documentation of the project. Google Drive will also be utilized for the purpose of working on shared documents in real time. All project data should exist on either GitHub, Google Drive, or both. The deployment server will not be the only place any data or system exists.

# **13. Training**

Further training in the C# programming language will need to be pursued by the current developers to ensure they follow standards and practices correctly. This is imperative to the quality of the project. In addition, research on proper SQL Server techniques and management will be required before implementing new functionalities that will rely on database connections. This is due to the fact that the developers that were assigned to database creation and management are no longer a part of the team.

# **14. Risk Management**

As stated previously, GitHub and Google Drive will host all code and documentation related to the project and its software quality assurance plan. The live project is hosted on a server that has been provided by Gannon ITS. Everything relating to the project that is on the server will be backed up on a corresponding repository. Developers will also have a current project version on their personal machines. These four hosting locations will help ensure that little to no documentation or code will be lost in the event of a form of catastrophe.

# **15. Glossary**

This section shall contain a glossary of terms unique to the SQA plan.

# **16. SQAP Change Procedure and History**

All changes to this document shall be noted in the “Revision History” section of this document, including a new version number, reason for change, and the name of the person that made changes.