Software Version Description

for

Scholar Station

**Version 1.0 approved**

**Prepared by Kenneth Dale**

**Ryan Knaggs**

**Cole Peaden**

**Joshua Shinde**

**University of West Florida**

**March 23, 2018**

**Table of Contents**

**Table of Contents 2**

**Revision History 2**

**1.** **Introduction 3**

1.1 System Overview 3

1.2 Version Overview 3

1.3 Team Assignments 3

1.4 Document Conventions 3

1.5 References 3

**2.** **Inventory of Materials 3**

2.1 Documents Released 3

2.2 Executable Media Released 3

2.3 Software Projects\Assemblies Released 4

2.4 Test Projects\Scripts Released 4

**3.** **Current Design 4**

3.1 Current Software Layers 4

3.2 Major Software Interfaces 5

3.3 Incorporating Design Patterns 5

3.4 Data Schema 5

3.5 Violations to Design Principles 6

**4.** **Remaining Work 6**

4.1 Remaining Software Features 6

4.2 Known Issues and Bugs 6

**Appendix A: Glossary 7**

**Appendix B: GitHub Information 7**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
| Joshua Shinde | 03/24/18 | Section 3.1, 4.1, appendix B. Reason for these changes was to add more of a description or fill in the section as there was no information. | 1.0 |

# Introduction

## System Overview

This software version document applies to the current version of Scholar Station. Scholar Station is a system students can use to receive and/or offer tutoring to assist in their academic pursuits. Students and tutors will leave feedback regarding these tutoring sessions and provide information regarding the areas they are struggling in, what was done to help them in these areas and the final outcome of the session. Ultimately, this information will be used by professors and other educational institution personnel to gauge student’s progress and the effectiveness of their curricula.

## Version Overview

Scholar Station v1.0 is the initial release of the software.

## Team Assignments

Kenneth Dale – Database creation, software design and documentation.

Ryan Knaggs – Software design and documentation.

Cole Peaden – Software design and documentation.

Joshua Shinde – Software design and documentation.

## Document Conventions

The content of this document does not contain jargon or abbreviations. It is straightforward and concise.

## References

# Inventory of Materials

## Documents Released

There is a user’s manual included in the release. It is titled “Scholar Station v1.0 User’s Manual.”

## Executable Media Released

The executable for scholar station is named “Scholar\_Station.exe.” To find this executable, navigate to Scholar\_Station/bin/debug folder and click on the executable.

## Software Projects\Assemblies Released

Scholar Station contains five projects. The names of these projects are DataAccessControler, DataAccsssControlerTest, Scholar\_Station, ScholarStationCL, and UnitFactoryUnitTest.

* DataAccessControler contains classes and interfaces that pertain to opening and closing database connections. In addition to opening and closing database connections, the DataAccessControler also contains interfaces that read and update the database.
* DataAccessControlerTest contains unit test that test the connectionControler classes and interfaces.
* Scholar\_Station contains classes that create GUI for the software system and all the logic that pertains to updating and communicating with the database.
* ScholarStationCL is a class library that contains interfaces for creating users. This project also contains a userfactory method that generates user objects.
* UnitFactoryUnitTest test the classes and interfaces within the ScholarStationCL.

## Test Projects\Scripts Released

This project utilizes mocks to test the database update feature. This test is the UPDate\_DBTest unit test within the DataAccessControlerTest project. This test uses a mock of IUpdate interface and returns a true boolean value if the update is successful. It also checks for failure and returns false if it fails.

In addition to the UPDate\_DBTest there is a DataReaderTest that determines if the datareader returns data from the database. This test passes a null string to the ExecuteQuery method to see if the dataReader returns a null datareader. If the data reader is null, this means that the passed null query string returns as expected. This test also test a true query string. This test will return a dataReader that contains data from the database. The test determines if the dataReader is null. If it is not, it means that the query worked and the test passes.

DBConnectionClosed checks if the database is closed. If the test returns true, it means that the connection is not open and that the test passes as expected.

OpenConnection checks if the database is ope. If the test returns true, it means that the connection is not open and that the test passes as expected.

# Current Design

## Current Software Layers

* Data Access Layer - This is the layer that contains our database. It provides simple access to the data stored within it.
* Business Logic Layer - This is the layer that contains our controllers and interfaces. This is how our data is created, stored, and manipulated.
* Presentation Layer - This is the layer that contains our user interfaces. This is what the user sees when utilizing and interacting with the software.

## Major Software Interfaces

The software system has four interfaces. IUserFactory is the name of the interfaces that deals with creating users for the software system. The method in this interface is create user. This method returns a user so that a new user can be instantiated to function within the software system. The DataAccessController project contains interfaces for opening and closing connections. It also contains interfaces for reading and updating the database. The IConnection interface has two methods. One of the methods is for opening a database connection and the other for closing the connection. The IRead interface allows interaction with the DatabaseControler to read from the database. The DataReader method in the IRead interface returns a SqlDataReader object and passes the contents of the database back to the calling object. The IUpdate interface contains a method that has no return type and is called ExecuteQueries. This method will open a database connection and write new contents to the database.

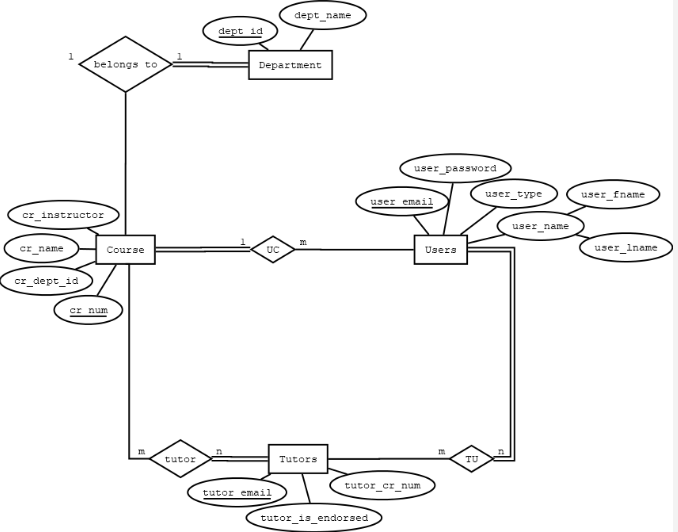
## Incorporating Design Patterns

* Null Object Pattern - To facilitate the development and testing of the software.
* Factory Pattern - To make the creation of a user simple and straightforward.

## Data Schema

The data schema for Scholar Station is a database which resides within SQL Server.

See below:



## Violations to Design Principles

# Remaining Work

## Remaining Software Features

* Passwords will be added to the login page.
* A user home page which will serve as a dashboard. This page will show all sessions the user is enrolled in, and will provide a link to a page with specific session details.
* A page specific to each session which will display session details. The user will be able to modify session details, cancel the session, and if the user is a tutor – close the session. Once the session is closed it will bring the user to the unimplemented feedback page.
* The feedback page will allow the tutor and student to leave feedback about the session attended. This feedback will be automatically emailed to the professor.
* Functionality to allow the user to create study groups.

## Known Issues and Bugs

The layout of the session information currently contains information which is not needed for the user.

**Appendix A: Glossary**

**Appendix B: GitHub Information**

GitHub Repository: Group4\_ScholarStation

Features currently being used:

* Pull requests