Technical Document

Table of Contents

[Technical Document 1](#_Toc34249055)

[Project definition. 1](#_Toc34249056)

[Application source location 1](#_Toc34249057)

[Presentation tier 2](#_Toc34249058)

[Business tier 2](#_Toc34249059)

[Data tier 3](#_Toc34249060)

[Technical Scope 3](#_Toc34249061)

[Technical Design 4](#_Toc34249062)

[Sequence diagram 4](#_Toc34249063)

[Use Case Diagrams 5](#_Toc34249064)

[Class diagram 6](#_Toc34249065)

[Technology used 6](#_Toc34249066)

# Project definition.

This project is a three-tier application that satisfied all requirements given in the related requirements document below.



## Application source location

<https://github.com/emyles1/AdvProgram>

The application integrates Visual studios UX forms, c# code and an SQL database to provide the administrative staff full access to Add, update and delete students from the DB.

The application will also enable Administrative staff to save student information in an XML format.

## Presentation tier

This application incorporates standard Forms UX interfaces from Visual studio, incorporating standard elements such as

* Textboxes
* Combo boxes
* Data grids
* Radio buttons
* Labels
* Buttons

A screenshot of a cell phone

Description automatically generated

## Business tier

The mid tier utilizes c# coding using the Visual studio SDK.

## Data tier

Backend utilizes Visual studios local DB Connection service to allow a local SQL DB within the SDK

SQL Initialisation script has been add below. It can also be found at the gibhub address above (Initiate.sql)



The script will require an existing DB to be created called StudentDB.

A Connection string will also have to be manually added to AppConfig.file.

Once all has been created and linked up. Login with the below credentials

Username: Admin

Password: Admin

# Technical Scope

The scope of this project was to create a 3 tier application to allow administration at a high level to;

* Login - Logout
* View Database history
* Add/Edit/Delete a student to the student DB
* Save a student to an XML file format.

These requirements were met using SQLConnections, SQLCommands utilizing stored procedures in SQL.

A logger was implemented to catch Administration interactions with the DB.

There was also a large effort to ensure the quality of the application is fit for purpose.

To achieve this there was a large effort to ensure all user cases were covered with try catch implementations, a Clear() class to ensure administration can’t change student details before transactions are complete and utilizing the UX to disable elements where not in use.

# Technical Design

## Sequence diagram

A close up of text on a white background

Description automatically generated

## Use Case Diagrams

A close up of a map

Description automatically generated

## Class diagram

A close up of text on a white background

Description automatically generated

# Technology used

* SQL Management server
* Microsoft Word
* Microsoft Excel