**Diploma in Software Engineering and Design**

# Assignment Cover Sheet

|  |  |
| --- | --- |
| **Course name:**  Diploma in Software Engineering and Design | **Student’s name:**  Kyle Harvey |
| **Module Name or Number:** Database Programming (15 credits) | |
| **Assignment title and/or number**: Database Assessment | |
| **Passing Criteria** | Need to score 50 or more marks to pass the assessment. **Total Marks: 100** |
| **Assessment weighting** | *12.5% of the overall programme.* |
| **Due date**: | **Date submitted**:  (late submissions incur 10% penalty, after 7 days late, the assessment will not be marked) |
| **Assessment conditions:** | This is a resource-based assessment. This means that you may have access to any relevant resources to assist you. This could include, for example, your learning materials, information on the Internet, and so on. However, all work must be your own with no assistance from any other person. |
| **Learning Outcomes** | * Install a database server * Create, open and close SQL connections * Demonstrate understanding of SQL injection * Perform create, read, update and delete operations in the database * Design a GUI to work with the database |

|  |  |
| --- | --- |
| **Requirement** | **Completed** |
| Database | ✓ |
| User interface | ✓ |
| Functionality | ✓ |
| Coding | ✓ |
| Testing | ✓ |
| Assessment Sheet | ✓ |

**Assignment**

**Checklist:**

# Disclaimer of Plagiarism and Collusion

I declare that, to the best of my knowledge, this assessment is my own work, and has not been copied from any other student's work or from any other source.

Enter your name here to indicate you agree to the above statement.

|  |
| --- |
| Kyle Harvey |

**C# Entity Database Assessment**

**Purpose**:

This assessment tests your programming and database skills. By completing this assessment, you will show that you can identify a business requirement and translate it into a software component. You will need to identify the requirement, prepare a database design and create a software component to work with the database.

**Requirements**:

1. Identify the business scenario and project requirement that you want to solve.
2. Discuss your project scenario with your tutor and get it approved
3. You need to use a database for the assessment and create a database design to work with your scenario
4. You will use Entity Framework to connect the database and perform CRUD operations
5. Code the software component using C#.net
6. Test your software by entering sample data
7. Your front-end could be Win Forms, WPF or UWP.

**Submission requirements**

1. A Software Requirement Specification (SRS) document which will include the purpose, scope and design of your project. Have a look at some of the links below for more details on what to include in a SRS document. https://en.wikipedia.org/wiki/Software\_requirements\_specification www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc https://belitsoft.com/custom-application-development-services/softwarerequirements-specification-document-example-international-standard

1. This document completed wherever required
2. Visual Studio project files
3. Database design
4. Database scripts
5. GitHub link
6. Testing document / Testing Log
7. Demo and Presentation (your tutor will inform you about the presentation day and time)

# GitHub link:

https://github.com/SumOfAllN00bs/VaporArchive

**Minimum Requirements**

1. Your project should have at least 3 Forms / Windows or you could have one window with multiple tabs.
2. Your database design should include at least 2 tables with primary and foreign key relationship
3. It needs to be a substantial project in a real-world scenario
4. The project needs to be well tested and robust in operation

**Timeline**

|  |  |
| --- | --- |
| Week 1 | Project Topic Finalised with Tutor Requirement  Specification document completed  Database design finalised  Phase 1 Completed |
| Week 2 | Phase 2 Completed |
| Week 3 | Testing and Bug fixing  Quality Control  Presentations |

**Assessment Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **% of Grade** | **Excellent 100%** | **Adequate 80%** | **Poor 60%** | **Not Met 0%** |
| User Interface    20% | Simple easy to use intuitive UI, no errors spelling mistake and good colour schemes used | Minor errors with the UI, minor layout issues | Major UI errors making it hard to understand | Significant UI errors with no logical sense and program crash |
| Coding    40% | No errors, code uses the best approach in every case and follows the coding standards | Minor errors or repetition of code, coding and naming standards not followed in some occasions | Code uses poorly-chosen approaches in some places. Naming standards not followed in some places. | Many things in the code could have been accomplished in an easier,  faster, or otherwise better fashion. Poor naming and coding standards |
| Testing    10% | Program is well tested to identify and fix bugs and errors. No major bugs or defects in the program. Testing results matches the actual program. | Program is well tested to identify most of the bugs, but some bugs still exist.  Some test cases marked pass which are false | Program has a lot of major bugs and is not tested. Testing sheet incorrect or incompletely filled out. | Program is not at all tested and testing sheet is not filled. |
| Documentation 15 % | The SRS is well written and clearly explains what the Software is accomplishing and how. | The SRS is somewhat  useful in understanding the software. | The SRS is explaining some parts of the project but is incomplete. | The SRS is not submitted. |
| Presentation 15% | Nicely prepared and complete presentation of the software app. Good use of Presentation aids and proper explanation. Good Body language. | Good  presentation  and demo  explaining the software. Minor details missed out or not explained correctly. | Not much effort put in explaining your software. Incomplete or Unprepared while presenting. | No presentation done. |