Faculty of Science and Engineering

Referred Coursework – 2019/20 Academic Year

PLEASE NOTE: If you have been referred in the COURSEWORK element of this module, please complete this referred work.

Module Code:	ISAD251
Module Title:	Database Applications Development
Module Leader:	Dr Shirley Atkinson
School:	Engineering, Computing and Mathematics

DEADLINE FOR SUBMISSION: 3pm on Thursday 20th August 2020

SUBMISSION INSTRUCTIONS FOR CANDIDATES

Referred coursework must be submitted electronically via the referred coursework submission point on the Module DLE page by the published deadline.

If you have any queries on submission or in relation to the referred work, please contact the Module Leader, Dr Shirley Atkinson (shirley.atkinson@plymouth.ac.uk) in the first instance, or the Faculty Office on 01752 584584 immediately so any problems can be rectified.

PLEASE NOTE that we cannot accept work submitted via email.

If you require any part of this publication in larger print, or an alternative format, please contact:

Faculty of Science and Engineering

T: +44 (0) 1752 584584
F: +44 (0) 1752 584540
E: science.engineering@plymouth.ac.uk

This document provides the referral information for the Coursework element. This is an **individual** piece of work and accounts for the 50% coursework element of the module. You should ONLY complete this work if you have been referred in this element.

Important Dates and Deliverables

Element	Description	Deadline	Percentage
	Online Review demo with ML (Appointment to be booked by student)	Monday 4th August From 10am	N/A
C1/W1 1	Final Individual Portfolio. DLE Submission	Thursday 20th August 2020 @ 3pm	50%

Useful Links

The University provides support for student wellbeing via https://www.plymouth.ac.uk/about-us/teaching-and-learning/guidance-and-resources/student-support-services. Please pay particular attention to the following two sections:

- Referencing and Plagiarism policy. https://www.plymouth.ac.uk/student-life/your-studies/essential-information/regulations/referencing
- Extenuating circumstances. https://www.plymouth.ac.uk/student-life/your-studies/essential-information/exams/exam-rules-and-regulations/extenuating-circumstances

Instructions to Candidates

Your task is to develop a database driven application for a family to manage their appointments, time and deadlines. Your application must meet the requirements given below.

The application is expected to be a simple three tier application having a simple browser-based client, middleware coded in a server-side language of either PHP or C# and a database contained in a Microsoft SQL Server database.

Code must be in a GitHub repository and typically include a storyboard, design documentation, current source code files and a screencast/capture of the application in use.

All code must be version controlled and commented, all 3rd party assets and resources are to be formally credited.

Pre-Production

Before you begin creating your database application you must carry out the analysis and design. You are to use a combination of modelling and storyboarding. It is expected that Entity-Relationship modelling (as taught in ISAD156) and UML modelling will be used. For those who have not encountered UML modelling and Storyboarding before, there are learning materials provided in this module. By using UML modelling and the technique of storyboarding you will be able to map out exactly the functionality of your application and how your pages relate to each other. You must create an architectural diagram of how your layers interact, a class diagram of how your code interacts (UML) and the website structure (storyboard) showing all the pages on the site and indicating how they are linked together.

You have one opportunity to carry out a peer review/demo with the Module Staff where you share and discuss your modelling, storyboard concept and ideas. You must record what was reviewed, the constructive comments made. This will form the basis of a final submission where you outline how you responded to that feedback. Use the scheduler on the DLE to book your slot with the module leader for the online review demo appointment.

Production/Alpha

Having completed your planning and prototyping if necessary, you will implement a final iteration based on the formative feedback and peer review. You will start with a final Application Design Document, bug fixing and refining/adding features. After extensive testing and debugging you will provide a final version on your Github repository.

PORTFOLIO (C1W1) - DELIVERABLES

You must submit one PDF file to the **module DLE** page containing your *Application Design Document*. This document will contain links to the following:

- A custom GitHub prototype page for your application containing the following:
    - Linked YouTube video (of you using your database application)
    - Screenshots of the application
    - Application fact sheet

- Your GitHub repository containing the following:
    - All source code for your database application (including .SQL files)
    - A link to your video hosted on YouTube.

Documents must be uploaded by the deadlines shown on the DLE. Ensure you do not leave uploading to the last minute or you may face a penalty if the server upload speed is too slow. Double check the submitted file was correctly uploaded. A second late will see you capped at 40%. It is recommended you upload the day before, earlier if possible.

PORTFOLIO DELIVERABLES IN DETAIL

Application Design Document

- This must be in PDF format.
- You must provide a storyboard to illustrate your application.
- You must provide an Entity-Relationship diagram that illustrates how your database has been constructed.

- You must provide appropriate UML diagrams to illustrate the architecture of your application and how the server-side classes are constructed and their interaction.
- Document any settings and images.
- Provide evidence that you have tested your web application using the Web Accessbility Initiative (WAI) guidelines with a commentary about what you would change and why.
- Include a list of the feedback acquired from the Demo sessions indicating what you changed and what you would change if you had more time.
- Provide the URL's for your API, your hosted web pages and your GitHub repo.

GitHub prototype page

- YouTube video
    - Length must be 2-5 minutes long
    - 720p to 1080p resolution
    - Show only the highlights of the application (not you on your webcam)
    - Must have audio to clearly hear you describing how you created your application
- Screenshots must be high resolution (720p to 1080p)
    - Screenshots show only the application, not the code Editor
- Application fact sheet
    - Application description of at least 200 words
        - Include a thorough explanation of the application and how to use it
        - Contains no spelling mistakes
        - Good use of English (sentence structure, punctuation etc)
        - A list of up to 5 key features
- Evidence of testing on more than one browser
- Evidence of Web Accessibility testing and results

Source code

- Must be clearly in its own separate folder on GitHub repository
- Technologies in use must be only SQL, HTML, CSS, Javascript and either PHP or C#

  - No other server-side coding language is acceptable.
  - The interface code may contain Razor if required.
- Frameworks such as Bootstrap and JQuery are expected to be in use
  - Templates such as W3.css are acceptable
  - No extra marks are provided for creating your own CSS stylesheets
  - All pages for the application are expected to have the same look and feel.
- Web application operates smoothly without crashing
  - Extra marks awarded for innovative approaches in use
- Middleware must be a RESTful API written in the languages mentioned above.
  - A minimum of two resources must be exposed using the main HTTP verbs of GET, PUT, POST and DELETE.
- SQL scripts must be contained within a folder labelled SQL
  - This should be an exact export from your hosted database.

Hosting

- API should be hosted on server-folder allocated to you for this module. Any deviation from this must be agreed with the module leader by one week from receiving this brief. The URL for this must be provided.
- Database must be hosted on the allocated Microsoft SQL Server instance.
  - Database must be populated with enough data to illustrate the application working effectively.
- Web pages should be hosted on server-folder allocated to you for this module. Any deviation from this must be agreed with the module leader by one week from receiving this brief.
- You should provide the URLs within your application design document even if they have been allocated to you.

GIT repository

- The readme file for your repository is appropriated completed
  - Repo readme includes all additional resources (images, libraries etc) fully credited
- The file structure is laid out appropriately

- No previous versions of the application are present in the repository in a .zip or other compressed format
- Commits to the repository are appropriately commented
- Commits are in a consistent timely manner, at least once every week, not all just before a deadline.

- Marks are awarded for evidence of good coding standards, including:
    - Logical naming conventions for variables, methods, classes etc
    - Thorough commenting
    - Logical separation of code
    - Reuse of code

NB: If we cannot access your material from the date of submission, we may not be able to mark it!

Anonymous marking cannot be used for this assignment due to the review activities needing to be traceable and the requirement for a video podcast.

APPLICATION REQUIREMENTS

This is an application that will provide the ability for an individual child to enter deadline details along with a parent user to enter the details for all the family. However, only have appointments and deadlines, do not add further items or categories*

The following user stories provide the basic requirements:

As a parent I wish to add details of dentist appointments for all in my family

As a parent I wish to see details of upcoming appointments for all in my family

As a parent I wish to change the details of any appointments for all in my family

As a parent I wish to add notes to an appointment that has taken place

As a parent I wish to cancel appointments for any in my family

As a child I wish to enter details of deadlines I need to meet

As a child I wish to see what deadlines I have coming up

As a child I wish to amend my deadline to show that I have met it

As a child I wish to amend my deadline to move it

As a child I wish to delete my deadline

*This is not a full system, this is a minimum viable product (MVP) whereby only the user stories provided above are implemented. The following aspects are outside of the scope of the system:

- Other users. No other user stories for other users are required.
- Logging in is outside the scope of the system. You are not to spend time implementing a bespoke log in system.

Expectations

There are expectations that your application will illustrate a quality of design rather than quantity. Appropriate security and validation is expected along with appropriate design of the interfaces. The application should have a modern look and feel as some form of styling is expected to be applied. Please refer to the marking criteria for further indicators of where marks will be awarded.

MARKING

Marks will be awarded under the following categories:

LO	Description	Total Mark (%)
1	• SQL statements are evaluated for defining, manipulating and controlling the data. • Evidence is sought for where SQL is used and where it can be found. • SQL is examined to evaluate how it meets the requirements for the application. • Evidence is sought for appropriate security approaches • Evidence is sought within the peer review and video for understanding how SQL should be applied.	25
2	• The application is examined to determine if it runs as expected. • The application is examined to determine how it meets the needs of multiple users.	25

	• Evidence within the application is examined to determine how the requirements are met. • The design of application is examined to determine if contents are where expected. • Evidence is found for appropriate security design • Evidence is found for appropriate legal, social and ethical considerations. Evidence is sought within the peer review and video for understanding how design might evolve	
3	• The server-side code is examined for quality and to see if written appropriately. • The application is examined to check it is clearly dynamic – data is delivered from the database to the browser and can be seen to be manipulated • The video is examined to find evidence that the student can demonstrate a suitable understanding of the code they have written • Evidence is found for appropriate security implementation • Evidence is sought within the peer review and video for understanding how implementation might be improved	25
4	• The API is examined to determine that it is RESTful • The application is examined to determine the level to which it utilizes the API • The API is examined to determine the level to which it delivers JSON data for four main HTTP verbs : GET, PUT, POST, DELETE • The API is examined for code quality. • Evidence is found for appropriate security features. • Evidence is sought within the peer review and video for understanding how API might be improved	25

GRADE BOUNDARIES

The following table illustrates the expectations for the different grade boundaries.

Boundary	Source code	Video	SQL
Fail Does not meet learning outcomes.	Source code has major elements missing or substantial errors. Code does not work and would never have worked.	Video does not show the application running. Video is poorly presented and does not demonstrate an understanding of the subject matter.	SQL statements are not correct. SQL statements do not run when imported into Microsoft SQL Server database. Fatal flaws observed in structure

3rd (40-49) Ok – but only just	Source code is readable but only by an expert. All elements present (web interface, middleware and database) Code is disorganized – spaghetti code! No reuse in place and in some places there are errors.	Video shows a very basic application, does not illustrate any code and has very little depth. Video is not well presented and only shows a partial understanding of what has been created. Student rambles and is not really coherent.	SQL statements run and create tables in Microsoft SQL Server database. Structure of SQL statements indicate some flaws.
2:2 (50-59) Some good elements but doesn't do all it should.	Majority of code conforms to coding conventions. Code runs as expected. All elements present (web interface, middleware and database)	Video attempts to convey appropriate implementation. The video is only partially effective at illustrating code and meeting of requirements. There is some evidence of understanding of the code but no consideration of future work	SQL statements run appropriately. Expected aspects are contained within the SQL but are very basic.
2:1 (60-69) Does all it should, is good, but plays safe	Code easy to understand and read. Reuse evidence. Format of code appropriate and logical. Evidence of testing, accessibility and appropriate security features.	Video illustrates application covers the requirements as expected. Student is coherent and walkthrough in video is structured appropriately. Video demonstrates student understands code and how they put it together.	SQL well written. Expected aspects are contained within the SQL and meet the requirements fully.

1st (70+) Shows self-discipline self-led learning and wide reading. Is excellent.	Code well organized and easy to follow. Adheres to good practice expectations. Excellent coding structures in place, clear evidence of quality approaches, testing and security. Application design shows student has carried out substantial self-led learning. Application goes beyond what has been taught in module	Video well presented, clear and concise. All requirements illustrated in concise fashion but with enough code to demonstrate how application created. Video able to show student has clear understanding of issues with database connectivity and manipulation demonstrated. The interfaces are of a very high standard The quality of the work is outstanding with no significant flaws. It demonstrates a good level of competence and depth	SQL excellent and adheres to best practice. Substantial self-led learning demonstrated. No significant flaws and demonstrates very high level of competence and depth.

Peer Review Template

Peer Review carried out by: (name) Date:

The peer review should pick one of the user stories and attempt to use the application as that type of user carrying out that task. On completing using the application, you should ask the author to describe to you one aspect of the code implemented to run the task – eg the database structure or the server-side code.

Task conducted: (Please enter the user story)

Was the task easy to carry out? (If no or not really, please say why)

Did you encounter any errors? (If yes, please explain what)

Did the author have to explain how to use anything? (If yes, what?)

Did you gain any inspiration for your own practice? (If so what?)

What constructive advice would you give the author for presenting their work/code in future?