



School of Computing, Electrical and
Applied Technology

ISCG7420

Web Application Development

Assignment 1

Semester 1, 2023

Due Date: May 7th 2023 17:00PM

Total Marks: 100

Course Weighting: 60%

Learning outcomes covered in this assignment.

2. Discuss the tools used in providing web-based applications.
3. Design and implement a dynamic web application using a range of languages/technologies/tools.
4. Discuss and evaluate the available data access technologies.

Assignment information:

- This is an individual assignment.
- Read the given scenario and complete the two tasks.
- For Task 1 you will evaluate two programming languages and two data access technologies and present your findings within a count of 1000 words [+/- 10%], excluding reference list.
- For task 2 you will design and implement a Django web application. Firstly, you will identify the Django models, then create views and templates.

Assignment Submission:

- Upload your response for task 1 in word format to Moodle "assignment 1-task 1 submission link".
- For task 2 use the Assignment 1 directory of your class GitHub repository to store your task and add your lecturer as a collaborator. Periodically throughout the course your lecturer will clone your repository to check your progress. On the assignment due date (and for up to 72 hours afterwards) your lecturer will download the most recent version of your task 2. You are required to ensure that GitHub always has the latest version of your code.
- For task 2 if you forget to push your most recent changes by the due date, then your lecturer will use the latest version available on GitHub when the deadline is reached for marking. **Make sure you regularly update files. If you use and submit code that you didn't write, you will fail this assignment, and fail the course. Ask your lecturer for help long before the due date instead of plagiarising.**

Scenario

School of computing is upgrading their gradebook system. This system allows lecturers to enter student marks. Students can also view their marks in the gradebook.

Business Rules:

1. One semester runs one to many courses
2. One course is run in zero to many semesters
3. One course can be separated into one to many classes
4. One class can only run one course
5. One class can be taught by only one lecturer
6. One lecturer teaches zero to many classes
7. One class holds one to many student's enrollments
8. One enrollment fit in one class
9. One student enrolls to one to many courses.
10. One student enrollment belongs to one student.

Client requirements:

1. Administrator can create/update/delete/show semesters.
2. Administrator can create/update/delete/show courses.
3. Administrator can create/update/delete/show classes.
4. Administrator can create/update/delete/show lecturers.
5. Administrator can assign/remove/change/show a lecturer to a class.
6. Administrator can create/update/delete/show student
7. Administrator can enrol/remove/show student to classes.
8. Administrator can upload students from excel files to the website
9. The system can email students when their marks are ready
10. Lecturer can login to the gradebook
11. Lectures can enter students' marks in the gradebook
12. Students can login to the gradebook
13. Students can view their marks in the gradebook.

Note: there is no need to change password and there is no need for registration.

Task 1: Technology Evaluation**[16 Marks]**

Before you start web development (i.e., in this case, grade book system) you will do research and evaluate programming languages and database technologies that can be used for the development. Answer the following questions.

Questions

1. Evaluate two web programming languages that can be used for this development.
2. Evaluate two database technologies that can be used for this development.
3. Based on your evaluation of questions 1 and 2 recommend a web program language and data access technology that can be used for development.

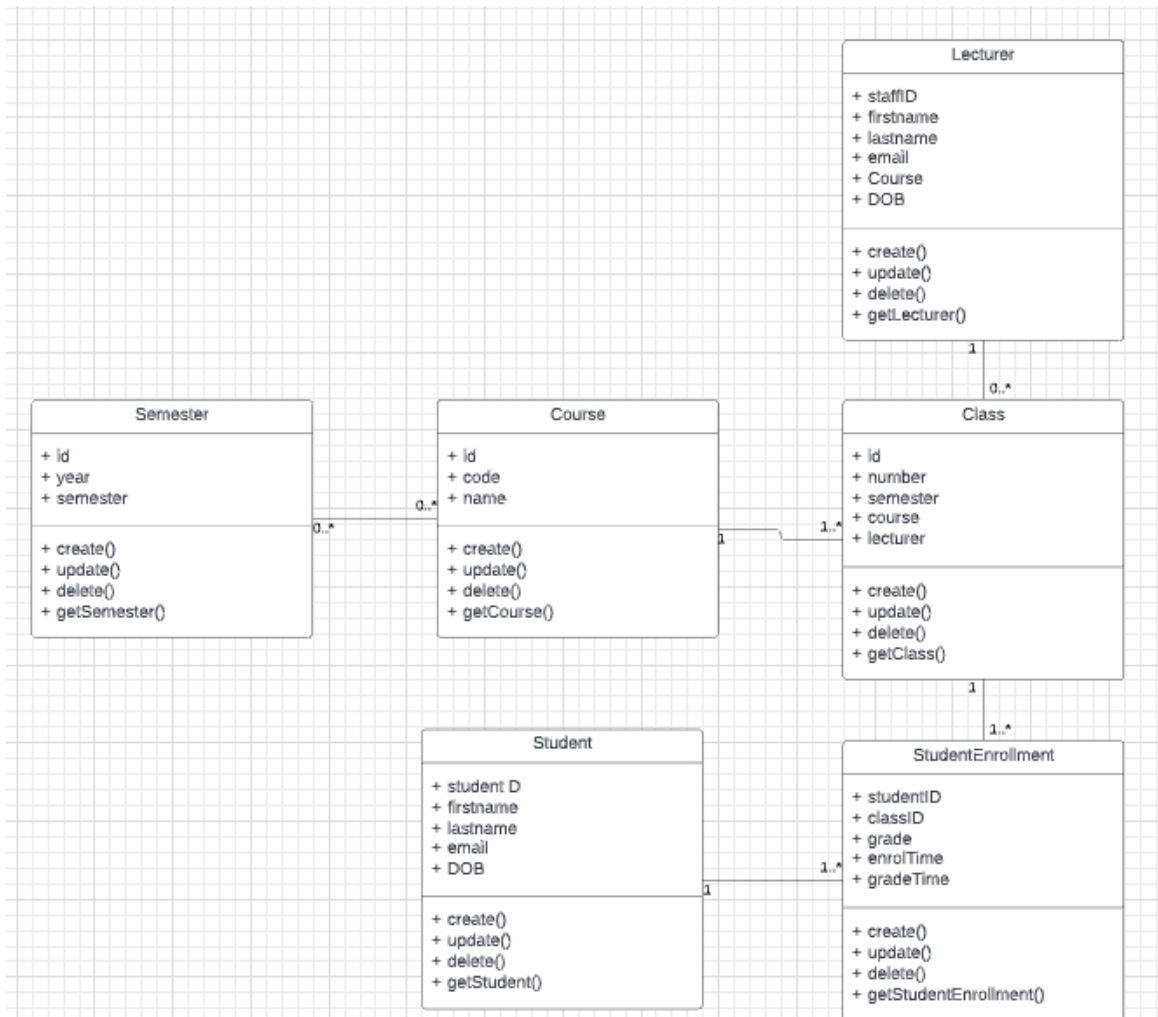
Task 2: Web Development**[84 Marks]**

To develop the grade book system, you need to design and implement a dynamic web application using Django, SQLite and/or other available web development tools. You will also host your web application on the Doc hyper server with available add-ons. You will upload your code to GitHub. The requirements for the application are given below:

Part A – Models**[36 marks]**

Follow the given class diagram to create models

Note: Many to many relationships are not shown in the class diagram, please figure out where you will use many to many relationships.

**Part B: Views****[34 marks]**

1. View for administrator to create/update/delete/show semesters
2. View for administrator to create/update/delete/show courses
3. View for administrator to create/update/delete/show classes
4. View for administrator to create/update/delete/show lecturers.
5. View for administrator to assign/remove/change/show a lecturer to a class
6. View for administrator to create/update/delete/show student
7. View for administrator to enrol/remove/show student to classes
8. View for administrator to upload students from excel files to the website
9. View for the system to email students when their marks are ready
10. View for lecturer to login to the gradebook
11. View for lectures to enter students' marks in the gradebook
12. View for student's to login to the gradebook
13. View for students to view their marks in the gradebook.

Part C: Templates:

[10 marks]

1. Template(s) for all administrator's functions.
2. Template(s) for lecturers' functions.
3. Template(s) for student's functions.
4. Authentications on all templates
5. Templates can be reused.

Part D: Deployment

[4 marks]

Requirements:

1. Setup Github for Django
2. Deploy your website to Dochyper.
 - a. Runtime.txt
 - b. Requirement.txt

ISCG7420
Web Application Development
Assessment 1 Mark Sheet

Student Names:

Task	Maximum marks	Your mark	Comment
1. Technology Evaluation	16		
2. Web Development			
a. Models	36		
b. Views	34		
c. Templates	10		
d. Deployment	4		
Total	100		

Late Submission of Assignments:

Assignments submitted after the due date and time without having received an extension through Affected Performance Consideration (APC) will be penalised according to the following:

- 10% of marks deducted if submitted within 24hrs of the deadline,
- 20% of marks deducted if submitted after 24hrs and up to 48hrs of the deadline,
- 30% of marks deducted if submitted after 48hrs and up to 72hrs of the deadline,
- No grade will be given for an assignment that is submitted more than 72hrs after the deadline.

Assistance to other Students:

Students themselves can be an excellent resource to assist the learning of fellow students, but there are issues that arise in assessments that relate to the type and amount of assistance given by students to other students. It is important to recognise what types of assistance are beneficial to another's learning and also what types of assistance are unacceptable in an assessment.

Beneficial Assistance:

- Study Groups
- Discussion
- Sharing Reading Material
- Reading the available online and library resources

Unacceptable Assistance:

- Working together on one copy of the assessment and submitting it as own work
- Giving another student your work
- Copying someone else's work, this includes work done by someone not on the course
- Changing or correcting another student's work
- Copying from books, the Internet etc. and submitting it as own work; anything taken directly from another source must be acknowledged correctly; show the source alongside the quotation
- Don't copy code from a website or video tutorial and pretend you made it or slightly change it. This will be an instant fail (0%).

Appendix 1

Sample Project Ideas (Simple versions):

1. Instagram clone
2. Facebook clone
3. Twitter clone
4. YouTube clone
5. Pinterest clone
6. Reddit clone

You must identify and construct the models required by most modern web applications. Example functionality is for example: making posts, uploading images, liking posts, comments, sending messages, adding a tag to a post, notifications, login/logout, etc.

If you choose to do something different you need to get approval from your lecturer.
Communicate with your lecturer during the course to ensure your application meets the required guidelines and marking schedule.

Appendix 2

Developer Journal Example:

===

March 23rd 2020 – Started @ 2:30PM

Going to work on Task #5, to try and implement saving user data to text files.

Estimate: 30 minutes

Got a bug, file isn't being created correctly.

5:15PM Still stuck on bug. Switching to Task #7 to add a button.

5:45PM Added a button, closed Task #7.

7PM: Fixed bug with file not being created. I was accidentally saving it to the wrong directory.

Future Suggestion: Make sure to always check which directory I save files to.

Actual time taken: ~4 hours.

7:30PM: Need to create an Invoice class which has save() method.

Estimate: 1hr

Stopped @ 9pm.

===

March 24th 2020 – 10AM

Continuing on Task #6 from yesterday – Creating class for Invoice

10:30AM: Finished class for Invoice (Task #6).

Actual time taken: ~2 hours.

10:45AM: Debug Issue #12 – Game collisions aren't working and object disappears off screen.

Estimate: 4 hours

1PM: While debugging game collisions I realized that I forgot to enable logging. I enabled logging and instantly saw why collisions weren't working. I wasn't checking if a collision occurred before moving the object.

Future Suggestion: When creating classes, start with a list of pseudo-code comments of things I need to remember to check. Also don't forget to enable logging when debugging.

Actual time taken: ~2 hours

2PM: Refactored / rewrote the `do_stuff()` function to be smaller and more clear.

4PM: Found a StackOverflow post for a possible solution to Task #9: <https://stackoverflow.com/questions/1077347/hello-world-in-python>

5PM: Solution didn't work. Out of time for this feature so will cut it and work on a different task.
Etc.. Write in your journal and work on your code daily.