SOFT355 DISTRIBUTED APPLICATION DEVELOPMENT

20 CREDIT MODULE / 90% COURSEWORK SUBMISSION / 10% IN-CLASS TEST

MODULE LEADER: DAVID WALKER

MODULE AIMS

This module explores the production of dynamic web applications with a particular focus on the web environment. Key elements such as object oriented and event-based scripting, asynchronous client-server communication and distributed content representation are explored through practical production. The production of working systems use frameworks such as HTML, CSS, JavaScript/JQuery and Node.js.

ASSESSED LEARNING OUTCOMES (ALO):

1. Apply principles of object oriented and event-based scripting as well as synchronous and asynchronous client-server communication.
2. Demonstrate an understanding of how application content is represented and communicated across the web and how this affects the user experience.
3. Design, implement and evaluate/test dynamic web-based applications.

OVERVIEW

SOFT355 – Distributed Application Development is a first semester module that provides you with experience of full-stack JavaScript development for web applications. You learn about writing client-side and server-side software using JavaScript as you use JQuery, Angluar.JS and Node.JS, as well as exploring how data is stored and communicated on the web.

The module assessment is twofold. You will sit an in-class test during week 13, which will be used to evaluate your JavaScript programming ability. The test contributes 10% of the final module mark, with the remaining 90% from a self-chosen individual project. Put simply, the project requires you to design and implement a distributed web application using the technologies and skills you have learned during the module. Though you are encouraged to start thinking about your application from the beginning of the module, the first deliverable is due in week 17 when you submit a project proposal. The purpose of the proposal is to ensure that the scope of the project is sufficient to achieve an excellent mark, and you will receive feedback in week 17. The final coursework submission is in week 24, after Christmas, and you are required to submit a report on the application and the code you have created. During that week you will be required to demonstrate your work to the module leader. In line with University regulations you will receive feedback on your coursework 20 working days following submission, in week 28.

MODULE DELIVERY

A lecture series with supporting workshop activities with topics including: client-side web development (HTML, CSS and JavaScript), JQuery, and AngularJS; persistence, including NoSQL and local storage technologies; peer-to-peer communication; and web-based software development and testing approaches.

Each lecture will be followed by a corresponding practical workshop in which students will apply the techniques they have learned in the lectures. Students are expected to self-select a workshop to attend, and should only attend one workshop per week.

Lectures:	Monday, 2pm-4pm Portland Square, Plymouth lecture theatre
Workshops:	Friday, 9am-11am Smeaton 104
	Friday, 11am-1pm Smeaton 104 / (Babbage 207 / Babbage 208 some weeks)
Duration:	12 weeks

Delivery staff:

Staff Member	Contact Details	Role details
David Walker	PSQ A322, david.walker@plymouth.ac.uk	Module Leader Lecturer, workshop series, general module queries.
Craig Banyard	craig.banyard@plymouth.ac.uk	Module Staff Workshop series.

Important Dates and Deliverables:

Element	Description	Deadline
T1	Mock in-class test released.	Week 12 Friday 18th October 2019
T1	In-class test.	Week 13 Friday 25th October 2019
C1	Coursework released.	Week 14 Friday 1st November 2019
T1	In-class test marks returned.	Week 15 Friday 8th November 2019
C1	Submission of coursework project proposal (D1) via the DLE.	Week 17 **3pm**, Monday 18th November 2019
C1	Feedback for coursework project proposal (D1) returned.	Week 17 Friday 22nd November 2019
C1	Coursework project final submission (D2) deadline, submission via the DLE.	Week 24 **3pm**, Wednesday 8th January 2020
C1	Coursework project demonstration (D3) – exact times and location TBC.	Week 24 Thursday 9th January 2020 Friday 10th January 2020
C1	Coursework project feedback and marks returned.	Week 28 Wednesday 5th February 2020

COURSEWORK – PROJECT AND DEMONSTRATION 90%

DESCRIPTION

This assignment contributes **90%** of the overall module mark for SOFT355 and is an **individual assignment**. Both the **proposal** and the **final submission** must be submitted to the DLE by the specified submission dates.

This is a negotiated project in which you must define the specific content you will produce and the process you will follow, within the guidelines given below. Examples of previous projects can be found on the DLE. Your project must satisfy **at least one** of the following requirements:

- The server should use a database to store information and pass it to the client through an API.
- WebSockets should be used to enable communication between two clients.

You must produce a working system, comprising a client constructed using dynamic web technologies (HTML, CSS and JavaScript) and a server using Node.js. The application must use JavaScript as the main development language both client-side and server-side. The system must be interactive, i.e., the user should be able to affect its behavior by interacting with it using a keyboard, mouse, or another suitable input device. The system must run on multiple computers, i.e., it must be distributed. The final product should reflect an effort of 80+ hours of dedicated work.

You must document the system, including its design and details of the implementation process you have followed, and provide a description of your DevOps pipeline. This could include your code repository, continuous information server setup, unit test, behavior tests, code analyses, usage metrics and usage analyses.

COURSEWORK DELIVERABLES

There are three main deliverables for this assignment. Deliverable D1, the project proposal, is a pass/fail deliverable, i.e., it does not affect the final module mark beyond deciding whether the project achieves a pass or a fail. Only deliverable D2, the final submission, and D3, the demonstration, will contribute to the final module mark beyond pass/fail.

D1 – Project Proposal

This is a brief summary of your planned application. It should explain what the application will do and which technologies you plan to use. The proposal should also explain how the application will satisfy each of the system requirements and how you plan to evaluate its performance.

The purpose of the proposal is to give you early feedback on the **suitability** and **feasibility** of your planned work in order to stop you spending lots of time on something that is inappropriate.

D2 – Final Submission

For this deliverable you must submit a single ZIP archive containing two main elements: your source code and a written report describing your project.

There is a **150MB limit on the final submission**. Please consider submitting your files from a computer on the University campus as submitting files of this size over an unreliable connection can be problematic. Please check your submitted files are correct by downloading them again and checking that they work. **You will receive a confirmation receipt by email when your work has been properly submitted** – if you do not receive this email then your work has not been submitted.

You must include the following elements in the final submission:

1. **Source Code:**
   This should contain all of the code you have written yourself and should, as far as possible, contain copies of the folders and files needed to run your application.

2. **Report:**
   The report must be a document of no more than 2,000 words. Please use screen shots and sketches to illustrate the functionality and UML diagrams (or appropriate alternatives) to illustrate the design. The report should explain:
   - Functionality (ca. 500 words with screenshots)
     - What does the application do?
     - How do the users interact with it?
     - What technologies were included?
   - Requirements (ca. 200 words with additional documents in appendices)
     - Who is the application aimed at?
     - What features were included and why?
   - Design (ca. 200 words with UML diagrams)
     - What is the system architecture (clients/servers/peers)?
     - How do the processes interact?
     - How are the data and code structured?
     - Why are these structures appropriate?
   - Testing (ca. 500 words)
     - What automated testing have you performed (e.g., unit testing)?
     - What usability testing have you performed?
     - Why is your chosen test strategy appropriate?
   - DevOps pipeline (ca. 400 words)
     - Describe your development environment.
     - Describe your continuous integration pipeline and how you used it.
   - Personal reflection (ca. 200 words)
     - What worked/didn't work well in terms of your work and the technologies you used?
     - What lessons would you take from this project into your next project.

D3 – Demonstration

The demonstration will take place in early January 2020 at a location and time that will be announced nearer the time. During the demonstration you must both demonstrate and explain how you implemented your application, including both the DevOps process and the application functionality. You will be asked detailed questions about how your code works.

PLAGIARISM AND ACADEMIC OFFENCES

You are expected to write your own code. Discussing ideas with other people is acceptable, but getting help to write your code is not acceptable. Failure to demonstrate a good understanding of your own code during the demonstration may lead to a significantly reduced mark on this assignment, or, in the worst case, action taken against you under the University's regulations on examination and assessment offences. Your attention is drawn to these regulations, available at https://www1.plymouth.ac.uk/essentialinfo/regulations/Pages/Plagiarism.aspx.

If you submit code files or libraries that you have not written yourself you must make this clear in your report. Submitting code that you have not written yourself without declaring it clearly in the report may lead to action taken against you under the University's regulations on examination and assessment offences.

Marking Rubric

Your work will be assessed according to the following mark scheme:

- Functional requirements and analysis (10%)
- Design and documentation (10%)
- Software quality and structure (40%)
- Testing (20%)
- Development and operations pipeline (20%)

Fail	> 40%	> 50%	> 60%	>70%
Insufficient problem analysis. Functional requirements are not described in sufficient detail. Little or no design work. It is not clear how the system will be built, or what it's major components are. Very little software constructed and what there is does not work. Little or no testing. Little or no version control. No attempt at an automated CI pipeline.	There are some vague functional requirements based on a basic outline of the problem. Some overview of the design but no detail. The architecture is not defined. Little or no use of diagrams to show the structure and operation of specific sections of the system. Most of the functionality has not been implemented. Major errors at runtime. WebSockets and/or a database have been attempted but incomplete. Some testing (either usability or unit). No automated CI pipeline. Some use of version control.	Reasonably detailed functional requirements but based on weak analysis. Some of the system's design is specified in detail but other areas are weak. Some diagrams included. Some requirements implemented but major omissions. Software contains major runtime errors. Either WebSockets or a database used. Reasonably complete testing (either unit or usability). Good use of version control. No automated CI pipeline.	Functional requirements well specified with some evidence of strong analysis. Design is mostly complete, further detail would enhance some areas. Most functionality and design specified with diagrams. Largely complete implementation of most requirements. Some runtime errors. WebSockets and database used, one of them weakly. Reasonably complete usability and unit testing. Version control used effectively. Some indication of test-driven development. CI pipeline has been attempted.	Functional requirements complete and based on strong analysis. System design complete and comprehensively described with use of appropriate diagrams. Implementation of all requirements is complete. WebSockets and a database are used, both work correctly. Reasonably complete usability and unit testing have been used in a test-driven development process. Other types of testing (e.g., integration, load…) have been used. Version control is used well. CI pipeline is well organised.

IN-CLASS TEST 10%

During week 13 you will sit an in-class test. The purpose of the assessment is to test your knowledge of HTML, CSS and JavaScript, and your ability to apply them in a practical environment. Following the test-driven development paradigm, you will be required to write code to fulfil a suite of unit tests with which you will be provided at the start of the in-class test.

The test is open book – you will have access to the internet. The in-class test is an **individual exercise** and has a duration of **50 minutes**. You will be assigned to a session, which will take place in one of the workshop sessions.

During the early workshops you will complete an exercise that is in the same format of the test. Two weeks in advance of the in-class test a mock test will be published.

MODULE INFORMATION

Please refer to all the lecture content & further study resources on the DLE.