Computer Science and Creative Technologies

Coursework or Assessment Specification

Module Details

Module Code	UFCF85-30-3
Module Title	Enterprise Systems Development
Module Leaders	Chris Simons, Mehmet Aydin
Module Tutors	Chris Simons, Mehmet Aydin, Dilshan Jayatilake, Martin Serpell
Year	2021 / 2022
Component/Element Number	B:CW1
Total number of assessments for this module	3
Weighting	42%
Element Description	Group-based demonstration of software development

Dates

Date issued to students	7 March 2022
Date to be returned to students	23 May 2022
Submission Date	5 May 2022
Submission Place	Blackboard
Submission Time	14:00 hours
Submission Notes	Compressed code to be submitted via Blackboard.

Feedback

Feedback provision will be	Verbal feedback will be delivered to the student following the demonstrations. Please see section 5.

Contents

Section 1: Overview of Assessment

This assignment assesses the following module learning outcomes:

1. *Understand the need for developmental frameworks for developing enterprise systems*
2. *Apply a state-of-the-art developmental framework e.g. Java Enterprise Edition to the team-based design and development of web-based applications*
3. *Discuss in detail the application of the notion of software architecture and software patterns in relation to analysis and design techniques for enterprise- scale software systems development*

The assignment is worth ***42%*** of the overall mark for the module.

This is a group assignment, which requires ***5 people*** to form a group for software development and demonstration.

Working on this assignment will help you to develop your technical development skills and well as your teamwork skills, which aims to improve your employability level by providing a practice of real working environment.

The specifications of this assignment are provided with a bundle of following files:

(1) UFCF85-30-3 2021_22 CW2 Spec.docx (main document)
(2) UWEFlix Case Study.docx
(3) Marking Matrix.docx
(4) Sprint Review Form.docx
(5) Peer Assessment Form.docx

You should make sure you downloaded the complete bundle and go through all files.

Section 2: Task Specification

You (as a group/team of 5 people) are asked to design and implement a web-based software system using the features and functionality of the examples from the course book, the practical classes, demonstrations in the lectures or other materials referred to within the scope of the module.

The assignment is based on the SmartCare Surgery case study, which is detailed in the enclosed file, called **UWEFlix Case Study.docx,** as part of the bundle. Please pay attention to the specifications and retrieve a list of requirements provided in the case.

You should design and build your software system using Python Django framework following the MVC design patterns, (MVT of Django implementation) using a Relational database of your choice for the backend.

As a team, you will be required to demonstrate and discuss your working system using VS Code IDE. Your team will be treated as a unit and any individual member may be required to demonstrate complete knowledge of the system you are presenting.

You are required to develop your ***web-based software system*** in last three sprints of Spring Term as follows:

Sprint 4: Task plan and schedule of delivery. This sprint covers ***Week 7*** and ***8***, mainly requires (1) working out with planning and scheduling the tasks, (2) setting up a DB among the alternatives, creating all tables with their relations (3) authentication for the majority of the user types. Your sprint review report for Sprint 4 should be uploaded on BB via File exchange at the end of ***Week 8***. The demo and assessment will be scheduled during practical of ***Week 9*** as a face-to-face review; all members are required to attend. This sprint weighs 10% of the overall mark.

Sprint 5: Completion of backend architecture and partial completion of front-end user interface. This requires, at least, half of the tasks are completed. These tasks are expected to be completion of all model classes, some templates including Forms and building `views.py`. This sprint covers ***Week 9*** and ***10***, its review is required at the end of Week 10, the demo and review will be assessed on ***Week 11*** fin the same way as previous sprint. This sprint weighs 30% of the overall mark.

Sprint 6: The group project is expected to be completed at the end of this sprint, which covers ***Week 11*** and ***12.*** Final and complete delivery of the application will be demonstrated and assessed on ***Week 13.*** It carries 60% of overall mark.

Section 3: Deliverables

You are required to submit your group work Sprint-by-Sprint as described above and with more detailed in Case Study document. The final version of the work must be submitted through the link to be provided on Blackboard VLE as a zipped project.

Sprint-wise Deliverables:

Sprint 4:

1. Sprint Review Form filled including Burndown chart, GitHub/Trello/Jira Kanban, task allocation/completion table.
2. Demonstrating completed tasks (10 mins).

Sprint 5:

1. Sprint Review Form filled including Burndown chart, GitHub/Trello/Jira Kanban, task allocation/completion table.
2. Demonstrating completed tasks (10 mins).

Sprint 6:

1. Sprint Review Form filled including Burndown chart, GitHub/Trello/Jira Kanban, task allocation/completion table.
2. Complete project zipped and uploaded on BB
3. Demonstrating the complete project (20 mins)
4. Peer Assessment Form (**Peer Assessment Form.docx**).
   This form should be filled and confidentially submitted to tutors by everybody, while other deliverables are to be in behalf of the groups.

Section 4: Marking Criteria

This group project requires multiple delivery stages (Sprints), where each stage is marked independently. Sprint 4, 5 and 6 carry 10%, 30% and 60%, respectively.

Marking tools are designed into marking matrixes as provided in the file, called "**Marking Matrix.docx**" within the bundle, where each sprint is considered in a separate matrix with more specific criteria and expectations. Please also see more general rules to follow for marking process detailed within Case Study document.

Marking Matrix document also includes a marking table for identifying individual contributions. Individual contributions are broken down following inputs from each member of the groups via Peer Assessment Form, separately provided with the file, called "**Peer Assessment Form.docx**" included in the bundle. The reported inputs by each group member is cross-checked with evidence provided via GitHub and other tools (to be) used.

You may note that 80% and beyond marks require outstanding excellence in the works, which are expected to exceed the requirements outlined in the Case Study document and should be genuinely exceptional.

Marking Process

As a team, you will be asked to demonstrate your developed software system in scheduled practical classes as applicable for each sprint where appropriate. For Sprint 6:

- You must download (from Blackboard) and unzip your project.
- You must run the SQL scripts provided to create and populate the required tables.

Your VS Code project and your DB server must be runnable on the standard FET configured laboratory machines (or the same configuration on your own laptop which you may like to use).

- It is your responsibility to attend scheduled classes – failure to demonstrate your system in class will be treated as a non-submission.
- All group members will be awarded the same mark as long as all group members agree that all contributed equally. Otherwise, your individual marks will be generated

based on your individual contributions identified based on collected and provided evidence. Everybody must confidentially fill and submit "**Peer Assessment Form**" for collecting inputs to determine individual contributions. A guiding scheme is included in the document called "**Marking Matrix**"

- Any group member failing to take part in the demonstration will be assessed as a non-submission and given zero marks.

The quality of your verbal expression in this demonstration is important – incoherent explanations will not achieve high marks. Please be advised that demonstrations last for a fixed-durations, so be prepared to concisely demonstrate and explain your system.

Some formalised Functional Requirements

The system-to-be is going to be tested and assessed with some test cases to make sure the following functional requirements are achieved: (test case data will be released in due-course):

1. The system accepts different user types and grant access to relevant functionalities subject to user privileges.
2. The Login/Logout functions work in line with Sessions, which expected to end after 20 non-active time. Terminated sessions logouts the user.
3. An ordinary customer does not need to login the system to book a seat and buy the ticket providing identification details. But a student or a club manager (representative) needs to login the system to gain access to booking facilities and buying tickets. The payment should be arranged to take off credit from student's or club manager's account, where each student and club manager should have credit account so as to be able to buy tickets. On the other hand, payment can be arranged based on some assumptions for ordinary customers.
4. There should be functionalities attached to each student and club manager users to top up their credits.
5. The duties/responsibilities and privileges of Cinema managers and workers can be retrieved/concluded from UWEFlix document, which includes advertising movies, arranging capacities, pricing the tickets, and applying discounts.
6. Adding new student users or club managers is within the responsibilities of cinema manager or worker, while the application should be made by candidate users.
7. Account manager looks after the cinema's finance, can observe daily transactions, generate periodical, (e.g., monthly or annually) financial reports.
8. Cancellation of bookings can be done upon request and subject to cinema manager's approval.
9. Changes in the prices and applying discounts should also be available for cinema managers or workers, while students and club managers should be able to request for further discounts.
10. Front-end design and further security measures beyond authentication will be considered as add-on.

You may assume the system will accept cookies. No threading/concurrency considerations are required. All data must be stored and retrieved from the DB of your choice.

Section 5: Feedback mechanisms

Feedback will verbally be provided immediately after the demonstration at each stage (Sprint). On your request, you can receive group-based and personalised feedback, written or verbally.