UNIVERSITY OF TECHNOLOGY, SYDNEY

41025 Introduction to Software Development

Project – Implementation & Testing

Assignment 2

Due Date: **Softcopy Due by Friday 31/05/2019 11:55 PM AEST**
Showcases in Week Commencing from 03/06/2019 -
During Related Workshop: See workshop schedule on UTS Online for exact dates/ time slots)

Submission: Each group will submit the following two items for the Online Movie Store:

Working Software: Working software application code will not be submitted via Turnitin. The individually implemented (modules or features) working software application code will be collated in a single project Zip including database. Submit ISD project working software code files (source and executable) with readme file (how to deploy and run the software) as a single Zip file in the UTSOnline Assignment 2/ "Working Software Code Submission" folder using your relevant workshop link **before 31/05/2019 11:55 PM AEST**. **You can submit the software only once. You must not make any changes once the software code is submitted. If you make any changes after the submission due date, then the late assignment rules will be applied.**

Report: Each group will submit a softcopy (Microsoft Word File or PDF) of the group report containing individual group member contributions. The individual student contributions or parts will be highlighted (with their names and student ids) and collated in a single group deliverable for submission and assessment (group submission but individual assessment). Each individual contribution will provide the (1) brief description of the assigned software application feature or module, (2) non-functional aspects, and (3) software testing results with executed tests and (4) defect log including (5) individual contribution logbook/timesheets via Turnitin **before 31/05/2019 11:55 PM AEST**. Use the Project - Assessment item 2 Turnitin link (View/ Complete) in the UTSOnline Assignment 2 folder for submitting your assignment.

Your both software code and report submission files title/name must follow the following naming pattern.

Your workshop activity number–group id

For instance, if your Wrk1 activity number is 02 (see timetable for your activity) and group id is G1 then your submission file title/ name must be worded as 02-G1. From each group only one student (project leader) should submit the assignment on the behalf of the whole group. You do not need to put the student ids of all the group members on the file title/ name. You must check Turnitin report and ensure that your work does not contain plagiarism.

You may submit your report to Turnitin many times before the submission due date. Final Turnitin reports can be used as evidence by the teaching staff in the event that plagiarism is suspected in an assignment and will be dealt as per University rules. **Do not allow anyone to copy your solution – this is considered misconduct; all miscreants will receive a mark of 0, at best for the assignment and will be dealt as per University rules**. You may be required to provide the hard or soft copy of the assignment anytime during the semester.

Marks: 50%

Word Limits: There is no word limit. It is not an essay. Therefore, it is not about the number of words or pages. Assignments in this subject are looking for quality and to-the-point professional work excluding unnecessary information or brain dump. You can express yourself in 20 or 200 pages, it is up to you. No student will be advantaged or disadvantaged by being less or more words or pages. **Focus on quality and not on the number of pages and words.**

Method: The assignment will be done in a group (preferably in the same workshop and same group as for assignment 1). Group size should be limited to **4 and no more than 6 students** (enrolment numbers and situation-specific circumstance will dictate the actual size of the groups). Groups were formed for assignment 1 (during Weeks 1-4) and for any reason(s) you want to change the group for assignment 2, then it is solely your responsibility to make other arrangements and find another alternative group who is willing to accept you. You cannot change the group once you have started the assignment 2 (Week 7 onwards). **This is a group assignment, however each student in the group needs to implement (code) and test end-to-end (as per MVC architecture layers) a complete working (free of defects) software feature or module as their individual contribution.** If the overall software works but an individual's implemented software feature or module does not work according to the requirements, then that individual will receive zero mark for this assignment. This way other members in the groups will not be affected by the no or poor performance of an individual. You must respect other students in the same group, different groups and teaching staff. **If you have any group issues, then you must inform your workshop tutor as soon as possible and well before (at least 1 week or earlier) the assignment submission or due date. Group assignment issues reported on or after the assignment submission date may not be considered. There will be zero tolerance for any academic and non-academic misconduct**. **See University Rules, Subject Outline and Academic Misconduct section of this brief for details**.

Objectives: Subject objectives: 1, 2, 3,4 and 5

1. Investigate and solve software development problems with minimal supervision.
2. Determine and balance the competing goals of software development activities within their constraints

3. Plan and manage a software development task to create, modify or extend a software feature or function to completion within the task constraints.
4. Apply sound software engineering practices to successfully create, modify or extend a software feature or function.
5. Communicate clearly software and task information to interested stakeholders

Type: Project

Groupwork: Group, individually assessed

Criteria: The assignment will be individually assessed based on the following criteria.

Criteria Items	Objectives	Weight
Working software application (35 Marks)	1,2,3,4	70%
Tests of the solution (10 Marks)	1,2,3,4	20%
Overall quality, presentation (5 Marks)	5	10%
Total	-	100%

There are no group marks. This means the individual marks for the Assignment 2 shall be based on individual contributions in all three criteria items across all the layers of the MVC architecture as a full stack developer. **Please also note that there will be no negotiation on a wrong answer**. An individual's mark for this group work assessment shall be computed as:

Individual Student Contribution & Mark = Working software application [assigned feature] (all the layers of MVC) + Tests of the solution [assigned feature] + Overall quality, presentation [assigned feature]

Task: You are required to develop a web software application for Online Movie Store (OMS) that allows users (registered and anonymous) to search and purchase available movies. OMS stock and transactions records are stored into a database. The OMS should be developed using agile practices and following and MVC architecture designed and planned in assignment 1. This assessment task will require a team of 4-6 students to produce, submit and **present a group report** (comprises of individual contributions), **small working software application** (comprises of individually implemented and tested software features or modules) and **individual contribution logbooks/ timesheets** for release 1 (see **minimum viable product section**) for Online Movie Store. Based on the plan, software requirements, architecture and design (submitted for Assessment Items 1) for release 1 (**minimum viable product**), each individual student in the group shall:

- **Working Software Application**: Each student in a group will implement and test the assigned (as agreed between the group members and approved by the tutor as a product owner) feature or module of the small software application; and

- **Report:** Each student in a group will provide the brief description of the assigned software application feature or module, non-functional aspects, and software testing results with executed tests and defect log.

The individual student contributions or parts will be collated in a group deliverable for submission and assessment (group submission but individual assessment). The deliverables of this assessment task also include a compulsory oral/visual presentation (no PowerPoint slides) of the individually implemented working software application during the scheduled assignment assessment or review session (showcase), individual contribution logbooks/ timesheets and working software code implemented – as per Subject Weekly Schedule. **Any whole team or individual student who failed to appear and present in these compulsory assignment assessment and review sessions (Showcase) will receive zero (0) as a final individual mark.** Students may choose to work in a lab or from home. Each ISD project team needs to nominate a project manager/ lead who will submit the assignment 2 (software and report) on the behalf of the whole group or team.

Summary

Students may choose to work in a lab or from home. Each group shall explore different ways of ensuring quality outcomes through the agile development and testing approach. This can be supported through a set of software development and testing tools. Please note that the work done in the Assignments 1 is a starting point for students to produce the working software and report in Assignment 2. **Working software must be developed and tested for release 1 of Online Movie Store (minimum viable product)**. Students must get feedback on their work-in-progress project from their tutors (product owners) during the workshop sessions before formal submission.

Note: It is recommended that the essential functionality of the assignment will be implemented using the web technologies and techniques taught in this subject (e.g. Java, JSP, JDBC, Java DB). You can use other technologies (JS, AngularJS, NodeJS, REACT, .NET, etc.) of your choice.

Minimum Viable Product

You have already implemented the index or landing page (home page) of the Online Movie Store (OMS) application in assignment 1 (prototype). It was the starting point of the web application and provided the options of login and register to users (without the program logic). It should provide further implementations, pages and links for other features (see table below) with appropriate navigation between pages (view), controller and model including database tables and sample data. In consultation with your tutor during Release 1/Iteration 0 (Week 7)/ Assignment 1 Showcase, you should confirm/ finalise the features from the following table (e.g. 1 feature per team member) for the minimum viable product, assign the individual feature to the team member and align the user stories captured in assignment 1. **Features 01, 02, 03 and 04 are mandatory features for the minimum viable product (release 1).** If you are 4 people in a group, then you will include these mandatory 4 features for the implementation and testing for release 1. For 6 people in a group, you will select these mandatory 4 features plus any of the other 2 features of your choice from the list to ensure that each student in a group has the responsibility to implement an individual end to end feature using MVC. Special circumstances or group size (<3) may change the group work in consultation with the tutor and subject coordinator. Professional software development and learning requires team effort.

Online Movie Store (OMS) Key Features Table (General Description: CRUD)

ID	Features	Create	Read	Update	Delete	Responsible
01	Online User Registration [MVC]	User can sign up for OMS (full name, email [as a user name], password, phone)	User can view their registration detail.	User can update their registration details.	User can cancel their registration.	e.g. put a name of the team member for the end to end delivery of this feature (MVC).
02	User Access Management [MVC]	A registered user access logs (user id, login date/time, logout date/time) are stored in the database.	A registered user can login and logout from the OMS. User can list (view) their access logs and search the log records based on the date.	User cannot update their user access logs.	User can delete their user access logs.	e.g. put a name of the team member for the end to end implementation of this feature (MVC).
03	Movie Catalogue Management [MVC]	User can create the movie details in the database. i.e. movie title, genre, price per copy, number of copies.	User can list the movie records. Search the movies based on the movie title and genre.	User can update the saved movie records.	User can delete the saved movie record.	
04	Order Management [MVC]	User (registered or anonymous) can create (save, submit) an order for a movie(s).	User can view their saved order details, order history list, and search the orders based on the order number and date.	User can update their saved order before final submission.	User can cancel their saved order before final submission.	
05	Payment Management [MVC]	User can create payment details (payment id, payment method, amount, date) for their movie order (order id).	User can view their saved order payment details, payment history list and search the specific payments based on the payment id and date.	User can update their saved payment details record before order submission.	User can delete their saved payment details record before order submission.	
06	Shipment Management [MVC]	User can create shipment details (e.g. shipment id, shipment method, date and address) for their movie order (order id).	User can view their saved order shipment details, history list and search the specific shipments based on the shipment id and date.	User can update their saved shipment details record before order finalisation.	User can update their saved shipment details record before order finalisation.	
07	User Management [MVC]	Admin staff can create users.	Admin staff can view the created user record, users list and search the users based on their full name and phone number.	Admin staff can update the user details and activate/ deactivate their status.	Admin staff can delete users.	
08	Customer Information Management [MVC]	Admin staff can create customer records (customer id, name, email, type, address).	Admin staff can view the customer record, customer list and search the customer based on their name and type [company or individual].	Admin staff can update the customer details and activate/ deactivate their status.	Admin staff can delete customers.	
09	Staff Information Management [MVC]	Admin staff can create staff records (staff id, name, email, position, address).	Admin staff can view the created staff record, staff list and search the staff based on their name and position [e.g. salesperson, manager].	Admin staff can update the staff details and activate/ deactivate their status.	Admin staff can delete staff.	

Additional Guiding Points:

1. Index page (home page) is the starting point of the web application and should provide the options of login and register to users and links to other feature pages (e.g. Movie

Catalogue, Order). Allow users to logout (to index page) or go back to the Main page from anywhere.

2. If a user cancels an order, then the order status is set to “cancelled”. However, the cancelled orders are still stored in database.
3. If a user cancels their account, all submitted orders made by this user should be automatically cancelled and orders details should be saved in database with their status marked as “cancelled”.
4. Once a user places an order, the number of copies of the purchased movie should decrease. If a user cancels the order then the number of copies purchased should be added back to the movies copies in database.
5. If a movie has zero copies, then users should not be able to purchase it.
6. As you add more pages to your web application make them available from home page or relevant page.
7. You must populate your database tables with sample data such as users, movies, orders etc. At least 20 records in each table. You can directly add these sample records through database management system interface.
8. Your web application should perform validation of inputs to prevent system crashes. In the case of your web application, you should display an appropriate error message if the user has inputted incorrect data, allowing them to re-enter the data.
9. The data validation should be server-side not client side (**do not use client side Java Script or CSS for data input validation**). This is for the purpose of server-oriented web development learning regardless of a good practice. Verify the input data against corresponding data stored in the database, where applicable.
10. Your user interface should be well thought out, providing a consistent look and feel on all pages, and providing useful navigation links. The user should be able to get to where he or she wants to go without ever having to click the browser's back button.
11. Your code should be well designed, commented and neatly formatted.

Assignment 2: Consolidated Working Software Deliverables & Report

ITEMS	Maximum Marks	Note
Cover Sheet, Project Title/ Header Page & Individual Student Contribution Page	**-**	**Cover Sheet & Project Header Page:** Sign, scan and embed FEIT declaration of originality cover sheet containing correct group name, student id, name and signatures in the report just before the project title/header page. **Individual Student Contribution:** At least 1 service/feature per team member covering all the layers of MVC (including database) for the report and working software for release 1 (minimum viable product) of Online Movie Store. Following are the features to choose from for the minimum viable product for release 1 (**see Minimum Viable Product Section**):

ITEMS	Maximum Marks	Note
		- Online User Registration - User Access Management (Login/Logout/ Access Logs) - Movie Catalogue Management - Order Management - Payment Management - Shipment Management - User Management - Customer Information Management - Staff Information Management **Roles:** - Registered user - Anonymous user - Staff Each service or feature can be linked to several user stories. Prepare and include the individual student contribution page with the following information **(see Minimum Viable Product Section):** • Student ID • Name • Assigned Feature/Service, • Short Description of Individual Contribution If you do not include these pages then assignment will not be marked, and you may receive zero for the whole assignment.
1. Working Software Application	**35**	**Working Software Code (MVC)**
View	(10)	Working user interface View pages. The View pages should have following items: 1. Meet feature/ user stories/ requirements (2 Marks) 2. Clear and consistent page layout, look and feel (2 Marks) 3. Easy to navigate between pages and main index page (navigation links) (2 Marks) 4. Provide validation (server side) of inputs (data) to prevent system crashes (2 Marks) 5. Provide an appropriate error

ITEMS	Maximum Marks	Note
		message if the user has inputted incorrect data, allowing them to re-enter the data (2 Marks) **Note:** Do not use client-side Java Script or CSS for data input validation.
Controller	(10)	Working Controller(s) to control the data flow between the View and Model. Controller layer acts as an interface between View and Model. It receives requests from the View layer and processes them, including the necessary validations (**server-side validations**). The requests are further sent to Model layer for data processing, and once they are processed, the data is sent back to the Controller and then displayed on the View**.** The Controller should have following items: 1. Meet feature/user stories/ requirements (8 Marks) 2. Reusable code classes or packages (2 Marks)
Model	(15)	Working Model layer. This layer contains business logic of the system and represents the state of the application. It's independent of the View layer, the Controller fetches the data from the Model layer and sends it to the View layer. Model layer is also connected to the Database. The Model layer including the connected Database should have following items. 1. Meet feature/ user stories/ requirements (5 Marks) 2. Reusable Java Beans (2 Marks) 3. Data Access Object (DAO) & Database Connectivity (2 Marks) 4. Database Tables & Attributes (2 Marks) 5. Appropriate Table data constraints & relationships (2 Marks)

ITEMS	Maximum Marks	Note
		6. Sample data in the database – at least 20 records in each table (2 Marks)
2. Tests of the Solution	**10**	**Report**
Software Application Feature or Service or Module	(1)	Provide a brief functional description of the assigned software application feature or module and its mapping to user stories.
Non-functional Aspects	(1)	Provide a brief description of the non-functional aspects of the assigned software application feature or module 1.
Software Testing Results	(4)	Document 1 acceptance test criteria for each user story relevant to your feature or service from release 1. Record acceptance test case (linked to a user story), and test results in the test matrix (e.g. excel spread sheet or MS word table).
Defect Log	(4)	Keep a log of the failed acceptance test cases in a table or spread sheet. Defect log should have at least following items (you can include additional items). Defect ID (DI001) Defect Description (e.g. problem and action) Defect Date Test Case ID (e.g. Failed test case id) Tester Name (e.g. who reported the defect) Responsible (e.g. who will handle the defect) Status (e.g. identified, assigned, in progress, resolved, unresolved defects) Comments (any additional comments) Summary: total defects, % of resolved defects, % of in progress defects.
Appendices – Project timesheet	-	Each student to complete and submit the timesheet signed by their project lead. **Note:** Assuming each student in the ISD project team is working 8-10 hours per week for this project. **If a student does not submit timesheets,**

ITEMS	Maximum Marks	Note
		then he/she will receive zero for their project mark.
Appendices – Individual Contribution Logbooks	**-**	Include contents from the Individual Contribution Logbooks. Link your individual contribution to weeks and hours recorded in timesheet. The individual contribution logbook is mandatory for students to submit with each Assignment: Assessment Items (1-2) to receive individual project marks. **If a student does not submit this logbook, then he/she will receive zero for their project mark.**
3. Overall Quality, Presentation	**5**	Quality of visual/oral group report and software demonstration. You are not required to prepare and submit the presentation slides. Launch and present the report submitted via Turnitin. Present software from your laptop. 1. Report is clear and easy to follow (1 Mark) 2. Report is correct from language perspective (1 Mark) 3. Software demonstration is clear (1 Mark) 4. Software code is commented and neatly formatted (1 Mark) 5. Overall correctness of answers to the questions (1 Mark)
Total Maximum Marks	**50**	

Note: You must demonstrate (present) working software (from your laptop) during showcase and submit working software code before assignment 2 due date. If as a team member you were not present during the showcase for demonstration, questions and answers relevant to your software feature or service module then you will receive zero individual marks for this assignment **(0 out of 50).**

Note: Please note that if the software failed to compile or run during the presentation or you did not present the software during the scheduled showcase time then **Zero** mark will be given for this assignment.

You should regularly get feedback on the assignment tasks and deliverables from the coaches during the workshop sessions. Each student group should assume the role of a Software Provider Start-up Company for the ISD Project.

Assessment Feedback

Feedback on the marked assignments will be within 2 weeks after the assignment due or submission date.

Minimum Requirements

See subject outline for details. **NO conceded passes are to be granted due to University Policy.**

Referencing Standards

All material derived from other works must be acknowledged and referenced accordingly using the Harvard Referencing Style (see http://www.bell.uts.edu.au/referencing/harvard_system).

Late Penalty

See subject outline for late submission penalty, unless an extension has been approved by the subject coordinator.

Special Consideration

Special consideration, for late submission, must be arranged beforehand with the subject co-ordinator (**email: asif.gill@uts.edu.au**).

Please also see the UTS Special Consideration Process:
www.sau.uts.edu.au/assessment/consideration

Special Needs:

Students should email the subject coordinator as soon as possible (and prior to the assessment deadline) to make them aware of the impact on them meeting assessment component/requirements, and that they are seeking assistance through UTS Special Needs as detailed in Section 5.1.3 of Procedures for the Assessment of Coursework Subjects.

Academic Misconduct:

Please see the subject outline for plagiarism and academic integrity in conjunction with UTS policy and procedures for the assessment for coursework subjects.

Querying Marks/Grades and Final Results

See subject outline for details.