Solent University
Coursework Assessment Brief

Assessment Details

Module Title:	Object Oriented Design & Development
Module Code:	COM528 (COM504)
Module Leader:	Dr Craig Gallen
Level:	5
Assessment Title:	Group Mini project presentation
Assessment Number:	AE1
Assessment Type:	Presentation / Group Software Artifact
Restrictions on Time/Word Count:	5 minutes presentation with supporting slides
Consequence of not meeting time/word count limit:	There is no penalty for submitting below the time limit, but students should be aware that there is a risk they may not maximise their potential mark. Assignments should be presented appropriately in line with the restrictions stated above; if an assignment exceeds the limit this will be taken in account in the marks given using the assessment criteria shown.*
Individual/Group:	Individual Presentation (group project)
Assessment Weighting:	40%
Issue Date:	27th September 2021
Hand In Date:	Friday 19 November 2021 16:00 (Week 8)
Planned Feedback Date:	4 weeks after hand in.
Mode of Submission:	Powerpoint and Video/Audio presentation via SOL
Anonymous Marking	This assessment is exempt from anonymous marking.

Assessment Task

You will be assigned to a small group and given the joint task of designing and developing a small Java Object Orientated Application. Your design documentation should include:

- project plan and task assignment
- use cases and a list of features
- test plan matching the use cases
- UML class diagram
- UML robustness diagram
- Your jointly developed code and documentation should be hosted in a github repository with appropriate documentation as markdown text.
- Your group will be required to submit your code on SOL and demonstrate your application working.

You are required to submit on SOL a 5-minute multimedia presentation and accompanying PowerPoint charts which should include sections describing.

- the use case
- the design and key design decisions
- problems encountered and solutions found.
- your contribution to the project
- a reflection on what you learned through the exercise.

Since several of you will be working on the same components, you may share diagrams (class diagrams, sequence diagrams etc,) between your presentations. However, the main content of each presentation should be in your own words.

You should properly reference any relevant literature and/or relevant software documentation or tutorials you have used.

Development Task

You are asked to create a small web based point of sale application which allows you to enter credit card details and a cash amount for a transaction. The application should communicate using a ReST interface to a mock banking service which will be supplied for you to use on-line. The service will confirm or deny the transaction.

(a) Key Features

More details will be provided in class but the following are the key features of the application.

- The UI should be a web interface and may use css and javascript libraries of your choice.
- The identity of the device should be enterable when the device is configured by an administrator and the identity credentials stored securely in a 'propertys file' which is read on start-up.
- The device should record all successful and unsuccessful transactions locally in a log file. cvv codes must not be recorded.
- users should be able to
  - enter a new transaction
  - reverse a transaction (refund to card)
  - check credit card Lunn code
  - allow entry of card number, name, expiry date, cvv code (card pin verification is not required)
- You should be able to demonstrate that your application is interacting with the bank service and that money is being transferred between accounts correctly with each successful transaction.

(b) Implementation

- You are required to use the technologies introduced in class without departing significantly from design patterns discussed in class (except after prior discussion with the tutor). Do not make the project difficult to mark by going off on a tangent.
- Your implementation should use enterprise java technology for the back end and may use javascript/css within the web client. Please note that marking will concentrate on the functionality and not the 'prettiness' of your implementation.
- You may use the skeleton multi module maven project introduced in class to give your project structure.
- You should provide unit tests for each layer/module in your design.

- You should use a logging framework to help with debugging.
- You should document all your classes with Javadoc.
- You should capture and log all internal exceptions and return meaningful error messages for user generated exceptions. Your program must not crash unrecoverably.
- Your submitted code must compile even if it is not functionally correct or complete.

Learning Outcomes

This assessment will enable students to demonstrate in full or in part the learning outcomes identified in the Module descriptors.

Assessment criteria

S, F3 - F1	D3 - D1	C3 - C1	B3 - B1	A4 - A1
Group Project Implementation (This grade will be shared across the group) (1/3 weighting)				
Little or no implementation Little or no project documentation.	Demonstrates a Clearly attempted and just sufficient implementation. Limited project documentation.	Demonstrates a Basic implementation of some simple requirements. Sufficient project documentation.	Demonstrates a Mostly Working implementation of limited set of requirements. Good project documentation.	Demonstrates a fully working implementation of all complex requirements. Excellent project documentation.
Personal Depth of Knowledge and Understanding (1/3 weighting)				
Fails to show sufficient understanding of the topic or the relevance of the subject matter.	Satisfactory understanding of the application of fundamental subject matter, but lacking in breadth or depth.	Demonstrates good subject knowledge and understanding of the key concepts relevant to the application of the topic to software development practice.	Possesses detailed subject knowledge and understanding of a range of concepts relevant to the application of the topic to software development practice. Evidence of specialisation in depth.	Shows a comprehensive and detailed knowledge of the topic and how it is applied to software engineering. Has a deep understanding of the wider context of the topic.
Personal Presentation (1/3 weighting)				
Fails to present relevant material in accordance with guidelines.	Meets the guidelines for the presentation. Poor use of referenced material.	Provides a coherent framework for the subject in hand, communicates content effectively. Reasonably referenced.	Forms a coherent whole and is consistent in its reasoning and presentation of fact Good referencing.	No errors of reasoning or fact, extensive reference to appropriate literature, detailed and coherent presentation

Late Submissions

Students are reminded that:

i. If this assessment is submitted late i.e. within 5 working days of the submission deadline, the mark will be capped at 40% if a pass mark is achieved;
ii. If this assessment is submitted later than 5 working days after the submission deadline, the work will be regarded as a non-submission and will be awarded a zero;
iii. If this assessment is being submitted as a referred piece of work then it must be submitted by the deadline date; any Refer assessment submitted late will be regarded as a non-submission and will be awarded a zero.

https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/2o-assessment-principles-regulations-temporary-amendments-for-covid-19-contingency-plans.pdf

Extenuating Circumstances

The University's Extenuating Circumstances procedure is in place if there are genuine circumstances that may prevent a student submitting an assessment. If students are not 'fit to study', they can either request an extension to the submission deadline of 5 working days or they can request to submit the assessment at the next opportunity (Defer). In both instances students must submit an EC application with relevant evidence. If accepted by the EC Panel there will be no academic penalty for late submission or non-submission dependent on what is requested. Students are reminded that EC covers only short term issues (20 working days) and that if they experience longer term matters that impact on learning then they must contact the Student Hub for advice.

Please find a link to the EC policy below:

https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/2p-extenuating-circumstances.pdf

Academic Misconduct

Any submission must be students' own work and, where facts or ideas have been used from other sources, these sources must be appropriately referenced. The University's Academic Handbook includes the definitions of all practices that will be deemed to constitute academic misconduct. Students should check this link before submitting their work.

Procedures relating to student academic misconduct are given below:

https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/4l-student-academic-misconduct-procedure.pdf

Ethics Policy

The work being carried out by students must be in compliance with the Ethics Policy. Where there is an ethical issue, as specified within the Ethics Policy, then students will need an ethics release or an ethical approval prior to the start of the project.

The Ethics Policy is contained within Section 2S of the Academic Handbook:
https://staff.solent.ac.uk/official-documents/quality-management/academic-handbook/2s-solent-university-ethics-policy.pdf

Grade marking

The University uses a letter grade scale for the marking of assessments. Unless students have been specifically informed otherwise their marked assignment will be awarded a letter grade. More detailed information on grade marking and the grade scale can be found on the portal and in the Student Handbook.

https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/2o-annex-3-assessment-regulations-grade-marking-scale.docx

Guidance for online submission through Solent Online Learning (SOL)

http://learn.solent.ac.uk/onlinesubmission