# Southampton Solent University

# Coursework Assessment Brief

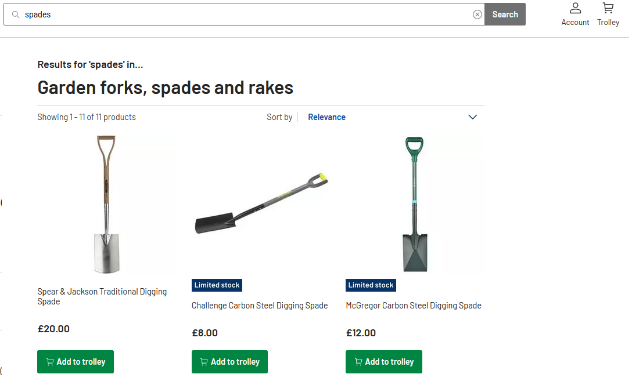
# Assessment Details

|  |  |
| --- | --- |
| Module Title: | Object Oriented Design and Development |
| Module Code: | COM528 / 504 |
| Module Leader: | Dr Craig Gallen |
| Level: | 5 |
| Assessment Title: | Individual Assignment – Catalogue Shopping Cart System |
| Assessment Number: | AE2 |
| Assessment Type: | Software Development |
| Restrictions on Time/Word Count: | N/A |
| Consequence of not meeting time/word count limit: | There is no penalty for submitting below the word/count 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 time/word count this will be taken in account in the marks given using the assessment criteria shown.\* |
| Individual/Group: | Individual |
| Assessment Weighting: | 60% |
| Issue Date: | 27th September 2021 |
| Hand In Date: | Monday 10th January 16:00 (Week 13) |
| Planned Feedback Date: | Within 4 weeks of submission |
| Mode of Submission: | on-line |
| Number of copies to be submitted: | 1 |
| Anonymous Marking | This assessment: is exempt from anonymous marking. |

# Assessment Task

This is an individual assignment which will build on the work you have done as a group in AE1. If useful, you may reuse any of the AE1 artifacts generate by your group but you must fork the code and add any improvements, extra features and documentation in a project hosted in your own Git repository.

**Case scenario (Shopping Cart System)**



You are to develop a simple web-based catalogue and shopping cart system which allows users to search for items, populate a shopping cart and purchase the items in the cart. The following scenario outlines the design requirements, but you are to develop a full design, feature list and test plan addressing as a minimum the following features. You may add additional features for usability as you see fit.

**Consider the following features in the development of the system:**

1. The system should support sessions with users in the following roles;

ANONYMOUS

* Any user who is viewing the site but has not logged in.

ADMINISTRATOR

* Able to add and remove and modify items from the catalogue.
* Able to able to view and modify users in the system.
* Able to able to view and modify orders in the system.

CUSTOMER

* Any user who has created an account and may have created orders.

DEACTIVATED

* A user whose account has been deactivated. This user cannot log into the system but details of previous orders will be retained for the administrator to see.

1. When the contents of a shopping cart are purchased, the shopping cart contents are converted into an order which can be tracked by the customer and marked as fulfilled by the administrator.
2. Each of the items in the completed order should be decremented from the stock.
3. All orders and transactions should be stored in the system. Customers should be able to review their own orders. Administrators should be able to review and search for all orders in the system.
4. When a user purchases an item, the system should interact with the bank service over a ReST interface to charge the user's credit card in a similar way to the group project AE1. A suitable Java ReST framework must be used to communicated with the bank. You may reuse code from AE1 for this functionality.
5. Any failed bank transactions should be logged and the user should be returned to the shopping cart in order to re-try the payment process.
6. The system should be backed by a database accessed through a suitable DAO/object relational mapping framework.
7. The user interface should be developed using a Java Web Technology (e.g. JSP/Tomcat) with supporting CSS and JavaScript as necessary.
8. The system should allow administrators to modify the contents of the product catalogue - adding and removing item types and updating the inventory of items.
9. The system should allow users to create accounts and enter their details (name, address, credit card details (but not cvv)).

**Design Process Requirements**

1. **Use Cases and Test plan**

You should document use cases for each actor in your system. These use cases should also be reflected in any integration and unit tests you create for your design. You should create a simple test plan (which can be manually performed) which illustrates the correct functioning of your use case. You will be asked to perform a selection of these tests as part of your presentation of your system.

1. **Model**

You should construct a model for your system documenting the core data types and interfaces you will need to implement for each component. Ideally you should use this model to generate the key interfaces and data elements but even if this is not possible, your programmatic elements should correspond closely to the model you have created.

1. **Robustness diagrams / Sequence Diagrams**

You should draw robustness diagrams and sequence diagrams to document how objects interact in your system including external entities such as the bank.

1. **Implementation**

* You are required to use the technologies introduced in class without departing significantly from design patterns discussed in class (except with 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.

1. **Packaging and Handing in**

* Submit your code and your report in a zip file on SOL BEFORE the submission date.
* All of your submitted code must be maintained in a project on Github. You MUST TAG your code using Git on Github as a submitted release and you must submit the tagged version on SOL.
* You should apply an appropriate open source licence and copyright to all of your code (e.g. Apache).
* You should include instructions for a user to build and run and use your submission.
* You should include documentation describing the key features of your solution.

1. **Report**

Your analysis, design, testing and code artefacts must be accompanied by a report. This should include discussion of the following (half a page to a page for each - I am not looking for reams and reams of text!):

* Decisions you made when drawing up your domain model and use-case texts.
* Any decisions made when drawing up your diagrams.
* Detail on places where your code did not match your design, and why.
* Rationale for your test strategy and test plan
* Critical evaluation of your code and/or design.

1. **Demonstration**

* A demonstration of your software will follow submission on an agreed date with the tutor
* Please note the demo represents 20% of the overall grade for this assessment and you will only be graded based on the tutor’s understanding of your software and report if absent for a demo.

# Assessment criteria

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A1-A4** | **B1-B3** | **C1-C3** | **D1-D3** | **F1-F3** |
| **Analysis and design (20%)** | Work fully complete; additional considerations beyond the basics have been made in your design. Analysis and design artefacts all consistent with each other. At least over 10 use-cases presented. | Work complete, analysis and design correct and artefacts all consistent with each other (a small number of inaccuracies or inconsistencies are permissible for a lower B). At least, up to 10 uses cases are presented | Work complete, diagrams predominantly correct and consistent with each other, but with a number of inaccuracies. Over 5 use-cases are presented. | Work mostly complete; significant inaccuracies and/or inconsistencies in your analysis and design. Up to 5 use-cases presented. | (F1) Some parts of the analysis and design completed, but others incomplete.  (Lower F) minimal effort. |
| **Implementation (20%)** | An implementation of all specified use cases which makes use of the more advanced implementation technologies covered in the unit.  Robust error handling and a user-friendly interface.  UI for with jsp’s for modifying catalogue and viewing invoices  Implementation matches design, or if not, the reasons for this are explained clearly in the write-up. | An implementation of all the specified use cases.  There may be room for improvement in your error handling.  Some evidence of use of the more advanced implementation technologies covered in the unit.  Implementation matches design, or if not, the reasons for this are explained clearly in the write-up. | At least three out of five use cases implemented. Little error handling.  Little evidence of use of the more advanced implementation technologies covered in the unit.  Implementation matches design, or if not, the reasons for this are explained clearly in the write-up. | At least 5 use cases implemented.  Implementation matches design, or if not, the reasons for this are explained clearly in the write-up. | A minimal effort; up to one use case successfully implemented. |
| **Testing**  **(20%)** | Comprehensive range of JUnit tests | A wide range of JUnit tests undertaken. There may be a small number of omissions | Significant number of JUnit tests undertaken but with a number of omissions | A small number of JUnit tests undertaken with significant omissions | Little or no testing undertaken |
| **Report (20%)** | Clear justifications of decisions made when drawing up your analysis and design artefacts including insightful comments. Considerations beyond the basics are made. Clear rationale for tests. | Clear justifications of decisions made when drawing up your analysis and design artefacts. Clear rationale for tests. | Largely clear justifications of decisions made when drawing up your analysis and design artefacts but unclear at times. Rationale for tests mostly clear. | Write-up clear and accurate in some places but unclear and/or inaccurate in others. A significant number of omissions. | Predominantly unclear and/or inaccurate write-up. Little understanding demonstrated. |
| **Test plan and Demo 20%** | Good test plan and demo of all features | Good test plan with a demo of most features | Good test plan with demo of some features | Demo of some parts of the solution. May be stubbed features | No working demo |

# Learning Outcomes

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

# Late Submissions

Students are reminded that:

1. 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;
2. 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;
3. 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>