# Southampton Solent University

# Coursework Assessment Brief

# Assessment Details

|  |  |
| --- | --- |
| Module Title: | Object Oriented Design and Development |
| Module Code: | QH0543 |
| Module Leader: | Hardik Bakari |
| Level: | 5 |
| Assessment Title: | Group Project – **PointsOfInterest** Console Application |
| Assessment Number: | AE1 |
| Assessment Type: | Software Development (Group) with individual presentation |
| Restrictions on Time/Word Count: | 5 min for the presentation (In class presentation during the Seminar) You **must upload** the slides to the SOL page while submitting your assignment |
| Consequence of not meeting time/word count limit: | There is no penalty for a presentation shorter than 5 minutes, but students should be aware that there is a risk they may not maximise their potential mark.  If the presentation exceeds 5 minutes, only the first 5 minutes will be assessed. |
| Individual/Group: | Group assessment, with individual components – **please carefully read the instructions within the assignment brief** |
| Assessment Weighting: | 40% |
| Issue Date: | 30 October 2023 |
| Hand In Date: | 31 January 2023 |
| Planned Feedback Date: | Within 20 working days 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

***HitTastic! - Stage One***

Your task in both assessments is to develop PointsOfInterest, a Java and JSP-based website, allowing users to search for, “like”, and comment on points of interest such as cities, historical sites, restaurants or pubs/bars.

In this first assessment, your task is to develop, as a group of two or three, a simple console-mode (no JSP required) object-oriented implementation of HitTastic! using Java. All input should be done via a simple console interface. You are not required to provide a “pretty” user interface; it just has to be usable.

There must be no more than three in the group under any circumstances. As soon as you have decided on your group, please inform the tutor by email. Any students not in a group by the end of Week 4 will be allocated to a group by the tutor.

You should include the following features. Note that at this stage you must not use a database. You should simply hard-code any data needed by the application (such as songs or users) as Java objects.

The module material from the first four weeks will be sufficient to complete this assessment.

***Group Member A:***

a) A user should be able to search for points of interest by **location** or by **type**.

b) A user should be able to “like” a point of interest.

c) A user should be able to add a comment on a given point of interest and modify a previous comment.

d) A user should be able to view their previous comments.

*As “add a point of interest” is a Group Member C task, Group Member A should simply hard-code a series of points of interest.*

***Group Member B:***

e) A user should be able to signup for an account.

f) A user should be able to login with a username and password. The system must be able to distinguish between regular users and administrators.

g) An admin user should be able to view a list of all users.

h) An admin user should be able to change the details of a current user (e.g. name, password) and delete a user.

***Group Member C (only applicable if a group of three; if a group of two, these requirements need not be implemented):***

i) An admin user should be able to see a list of all unauthorised comments made by users so far. The point of interest that the comment relates to must be visible in the information displayed.

j) An admin user should be able to search for comments by user.

k) An admin user should be able to add a new point of interest.

l) An admin user should be able to authorise and delete comments. Comments should be referenced by a unique ID.

*As “add a comment” is a Group Member A task, Group Member C should simply hard-code a series of comments.*

**How to do the assignment**

*Code*

Groups should divide the work up between members so that each member takes responsibility for one of the three groups of tasks above (Group Member A, B or C). If you are in a group of two, each member should implement either the Group Member A tasks or the Group Member B tasks – you should not implement the Group Member C tasks.

You must not put all your code inside a single main() method within a single class – you must create other appropriate classes and document them in your class diagram (below)!

You should consider *reusability* when writing your code: please see the material in weeks 2 and 3 which discusses this.

*Class diagram*

Each group member should then produce a class diagram for their individual tasks, showing classes needed, an initial guess at attributes and methods, and relationships between classes. Each member should also create a simple Java project in NetBeans to implement their own tasks. Focus on implementing a simple Java application. There is no need to use a database for this assignment – even to reach a high grade.

**Integration**

When each member of the group has got their requirements working, you should, as a group, integrate all the requirements to produce a working application. As part of this, ensure that only logged-in users can access the Group B admin functionality, and Group C functionality if applicable. You should discuss what changes you had to make to the code as comments within the code.

**Presentation**

Each individual student should produce a recorded multimedia presentation. This should should describe their individual implementation of their own tasks (length: 5 minutes – please do not exceed this, the assessor will stop listening after 5 minutes and any further content will not be graded). It should take the form of an explanation of your class diagram, including:

- an explanation of the role of each class,

- a summary of how the methods work technically,

- and an explanation of any interactions between the classes.

You should not include in this discussion the “main” class containing the console input and output, and you should not include the group integration process. The sound quality must be good – you **must** test this; the assessor will not attempt to listen to quiet or distorted audio, and such presentations will receive a mark of zero.

**What if I have to work individually?**

It is recognised that there may be problems working within a group, for example a group member may become ill and have to take a significant amount of time off university. If you have made every effort to work with your other group member(s) but it is not possible, you may work on this individually, but you must inform the tutor in writing, clearly explaining the situation (e.g. via email), at least one week in advance of the hand-in date.

Select the tasks for either Group Member A or Group Member B, and work on them individually. Then, produce a short one page report describing what steps you would have taken to integrate your work with the other set of tasks (from Group Member B or A), what problems might have arisen, and how you would have solved them.

**How you should hand-in the work**

1. A **single nominated member of the group** (you must inform the tutor who this is) should hand-in a ZIP file by the specified deadline, containing the code for the whole group project, written by all group members.

This must contain clear comments explaining how individual code was integrated to produce a working application. The comments must be consistent with what actually happened (I can check this via the individual code submissions); if they are not, the whole group will receive a mark of zero for the group integration.

The code should also identify the individual authors (or “group” for group-authored work) as comments. Any code with no apparent author(s) will not be marked.

2. In addition, **each individual student** should upload a ZIP file to SOL by the specified deadline, containing:

- your individual code, before integration (individual)

- your class diagram (individual)

3. Each individual student should also upload their presentation as a video/audio submission. Please do not put it in the ZIP file, otherwise it will not be marked. As I stated above, you must test the sound quality of the uploaded presentation, and this must be done after upload to SOL. If the sound quality is poor you will need to re-record and re-upload; quiet or distorted presentations will receive a mark of zero.

# Assessment criteria

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A1-A4** | **B1-B3** | **C1-C3** | **D1-D3** | **F1-F3** |
| **A. Class diagram (10%)** (individually marked) | Clear and complete class diagram with good object-oriented design | Class diagram shows object-oriented design with multiple classes to represent different entities in the system. Some small potential for improvement | Class diagram shows object-oriented design with multiple classes to represent different entities in the system. A number of omissions or unclear aspects | Class diagram shows object-oriented design with multiple classes to represent different entities in the system – but with a significant number of flaws or omissions. | Predominantly unclear and/or inaccurate class diagram |
| **B. Individual Implementation (40%)** (individually marked) | All individual tasks covered to a high standard.  Object-oriented solution, as for C-grade criteria. | The vast majority of individual tasks covered. There may be very occasional errors  Object-oriented solution, as for C-grade criteria. | Most tasks covered with a few omissions. Some errors may be present but the code should be runnable and testable  Object-oriented solution with multiple classes to represent different entities in the system. | A significant effort made on the tasks but with significant omissions. Some errors may be present, but the code should be runnable and testable  Mostly object-oriented solution with multiple classes to represent different entities in the system. | Code which does not run successfully. Minimal effort made  Insufficiently-object-oriented solution with an inadequate number of appropriate classes. |
| **C. Presentation (10%) -**  (individually marked) | Clear and detailed presentation describing all classes and methods used in technical detail. | Mostly clear and detailed presentation with occasional omissions or unclear sections | Mostly clear and detailed presentation with a number of omissions or unclear sections | Presentation partly complete but with significant omissions or unclear sections | Predominantly unclear and/or inaccurate presentation Little understanding demonstrated. |
| **D. Integration of Code, with comments (40%)** (group marked) | Clear and detailed comments on integration process. Complete application works seamlessly | Mostly clear and detailed comments with occasional omissions or unclear sections. Complete application works seamlessly | Mostly clear and detailed comments with a number of omissions or unclear sections. Complete, combined application mostly working, with some omissions | Comments partly complete but with significant omissions or unclear aspects. Application has successfully combined work of group members but there may be significant omissions | Predominantly unclear and/or inaccurate comments. Little or no attempt to integrate code. Little understanding demonstrated.  Comments do not match the actual changes to the code which occurred |

# Learning Outcomes

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

**Living CV**

As part of the University's Work Ready, Future Ready strategy, you will be expected to build a professional, Living CV as you successfully engage and pass each module of your degree.

The Living CV outputs evidenced on completion of this assessment are:

1. A simple group-developed object-oriented Java console application. This can be published to GitHub for others to view.

Please add these to your CV via the Living CV builder platform on Solent Futures Online [Solent Futures Online](https://eur03.safelinks.protection.outlook.com/?url=https%3A%2F%2Fsolentfutures.careercentre.me%2Fprogrammes%2F%3FprogrammeID%3DThzJ%252bRbk%252bQXoSlEaujPR0g%253d%253d&data=04|01|ian.harris@solent.ac.uk|f1bda34c4d564e82f6cb08da067fdf48|d684e4cd491a4577bf33546478d72e3c|0|0|637829443517919744|Unknown|TWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D|3000&sdata=ObCFbM3zY7CgU6SVNtitaq1udg0%2Bzlp1GuCAJ1y1utw%3D&reserved=0)

# Late Submissions

You are reminded that:

1. If this assessment is submitted late i.e. within 7 calendar 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 7 calendar 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.

[Assessment regulations](https://www.solent.ac.uk/about/documents/assessment-regulations.pdf)

# Extenuating Circumstances

The University’s Extenuating Circumstances (EC) procedure is in place if there are genuine short term exceptional circumstances that may prevent you submitting an assessment. If you are not 'fit to study’, you can either request an extension to the submission deadline of 7 calendar days or you can request to submit the assessment at the next opportunity, i.e. the resit period (as a Defer without capping of the grade). In both instances you must submit an EC application with relevant evidence. If accepted under the university regulations there will be no academic penalty for late submission or non-submission dependent on what is requested. You are reminded that EC covers only short term issues (20 working days) and that if you experience longer term matters that impact on your learning then you must contact the Student Hub for advice.

Please find a link to the EC policy below:

[Extenuating Circumstances](https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/2p-extenuating-circumstances.pdf)

# Academic Misconduct

Any submission must be your 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. You should check this link before submitting your work.

Procedures relating to student academic misconduct are given below:

[Academic Misconduct](https://students.solent.ac.uk/official-documents/quality-management/academic-handbook/4l-student-academic-misconduct-procedure.pdf)

**Ethics Policy**

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

The Ethics Policy is contained within Section 2S of the Academic Handbook:

[Ethics Policy](https://staff.solent.ac.uk/official-documents/quality-management/academic-handbook/2s-solent-university-ethics-policy.pdf)

**Grade marking**

The University uses an alpha numeric grade scale for the marking of assessments. Unless you have been specifically informed otherwise your marked assignment will be awarded a letter/number grade. More detailed information on grade marking and the grade scale can be found on the portal and in the Student Handbook.

[Grade Marking Scale](https://staff.solent.ac.uk/official-documents/quality-management/academic-handbook/2o-assessment-regulations-annex-1-grade-marking-scale.pdf)

**Guidance for online submission through Solent Online Learning (SOL)**

[Online Submission](http://learn.solent.ac.uk/onlinesubmission)