# Southampton Solent University

# Assessment Brief

# Assessment Details

|  |  |
| --- | --- |
| Unit Title: | Software Systems Development |
| Unit Code: | COM600/BCC608 |
| Unit Leader: | Dr Craig Gallen/Sahan Perera |
| Level: | FHEQ level 6 |
| Assessment Title: | Report 1 |
| Assessment Number: | AE2 |
| Assessment Type: | Report |
| Restrictions on Time/Length : | 2000 Words |
| Individual/Group: | Individual |
| Assessment Weighting: | 60% |
| 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. |
| Issue Date: | 1st September 2022 |
| Hand In Date: | 19th December 2022 |
| Planned Feedback Date: | 09th January 2023 |
| Mode of Submission: | Online |
| Number of copies to be submitted: | One |
| Anonymous Marking | This assessment is exempt from anonymous marking as it falls within an exempt category under the University’s Anonymous Marking Policy. |

# Assessment Task

This assignment takes the form of a software development exercise, the scale and complexity of which is sufficient for the student to apply the development principles that they have studied in a meaningful context.

# This will be a reflective report but will be backed up by evidence of your work throughout the unit. Its weighting is 60%. The supporting evidence i.e. the software development artefacts, including an executable, will be uploaded to Solent Online Learning along with your prepared reflective report.

# Assessment criteria

# See attached grid.

# Learning Outcomes

This assessment will enable students to demonstrate in full or in part the learning outcomes identified in the unit descriptors. The specific learning outcomes addressed by this assessment are as below.

# 

**Knowledge and Understanding**

K1 Explain and discuss industry-recognised architectural patterns, design patterns, code idioms and other heuristics relevant to the development of larger-scale software systems.

**Cognitive Skills**

C1 Synthesise the principles and patterns of software design in providing a solution to a software development or maintenance problem.

**Practical and Professional Skills**

P1 Apply ‘agile’ development approaches to the design, construction, testing and evaluation of a software solution.

P2 Assess the benefits of team-working as an approach to software development.

**Transferable and Key Skills**

T1 Research a given topic and discuss the outcomes.

**Introduction**

The professional development principles, ideas and tools that will be employed in the development exercise include:

* Using appropriate software design diagrams as part of the ICONIX process
* Create a working MVC in a suitable programming language
* Using GRASP and/or GoF design patterns within the development
* Using appropriate automated test tools for unit and/or acceptance testing
* Using appropriate management tools for both team-working and version control.

Students will therefore apply industry-recognised analytical, design, testing and project management techniques toward the development of a system.

The system is a **Demonstration Program** which acts as a prototype for an application to be used within a department of a Tourism organization that organizes tour packages.

The students will begin by working in small teams in a SCRUM type environment. Once a base level product and been designed, tested and accepted, the students can either continue working in teams, or can progress independently. (If you do work independently, you will still use most of the development artefacts, but you will not gain the experience of working in a development team.)

The assessment criteria focus on an individual’s report, either as a player engaged in a team, or as a sole developer. If, in the unlikely event you spend most of this assessment working by yourself you should address the “Team work” criteria by considering your perception of the benefits/issues of working as a member of a team in an agile software development environment.

The seminar sessions will introduce several software development concepts. Students will be encouraged to research these topics and to include them, as appropriate, into their activities; development topics will include SCRUM, the MVC pattern, the ICONIX process, test driven development, GRASP and GoF architectural patterns, other software design patterns, version control and content management.

**Activities**

This assignment requires you to *incrementally* and *iteratively* develop the Motorcycle Tour Demonstration Program following the SCRUM Project Management technique and using the ICONIX software development process.

**Assessment**

Assessment will be based upon the criteria set out in the attached grid which is derived from the FHEQ Level 6 assessment descriptors. You are advised to study these criteria in detail.

This is an individual assignment (regardless of whether you have worked as a member of a team, or independently).

Students will convey their own reasons and justifications for design and management decisions.

Students will present their own individual reflective elements when evaluating different designs and developmental techniques.

Where a student draws on external sources proper reference must be made according to standard University guidelines.

**Background of the Tourism organization**

The case study for this assignment is “A Demonstration Program which emulates some key activities within a department or section of a Tourism organization which organize tour plans/ tour packages and other activities related to tourism industry”.

The Tourism organization is likely to receive the following types of enquiries:

Organization related request:

Questions regarding packages

Questions regarding how to book a package

Legal requirements for overseas tourists.

Tour packages related requests:

Availability and Cost

Bookings

Confirmations

Cancellations

Providing feedback

The Tourism organization is likely to want to perform the following tasks:

View/Add/Update/Maintain hotel information

View/Add/Update/Maintain tour booking information

View/Add/Update/Maintain customer information

Gather and view customer feedback for each package.

Packages can be reserved by overseas and local customers.

A package is regarded as having the following attributes:

* A package can include multiple destinations and activities (hotels, camps, hiking, rail-hiking, etc.).
* Some options can be customizable based on customer preference.

Payment for package is dealt with separately by the organization and not part of the system. For insurance purposes all tourists must register.

You are encouraged to:

* + Document the overall human system using a Rich Picture from Soft Systems Methodology
  + Visualise your thoughts with radiant structures of Mind Maps to help to explore and gather ideas.
  + Use Mind Maps to help you form initial User Stories.
  + Each User Story should be developed on a Card (e.g. Jira Epic & Story if you use Jira). The card can also contain estimation and testing information.
  + Develop the details of each User Story by having a conversation with the Product Owner (tutor).
  + Develop acceptance tests that confirm each User Story is coded correctly.
  + Develop your Product Backlog based on your User Stories.

**Managing your product development using SCRUM**

Once you have developed your Product Backlog based on your User Stories, you identify the features for your next release. Use these to create your next Release Backlog.

Prioritise the features and estimate the amount of work involved for each feature.

Plan out several Sprints to implement this Release.

Each sprint should have a Work Breakdown Chart showing the components of the sprint and which components are delegated to whom.

Create a Burndown Chart and update it after each sprint. This will show your average rate of productivity and your estimated completion date. You can compare the actual velocity and projected completion date with what the team needs to do to complete on time.

All project management documentation and charts, whether created within your team or individually, where appropriate, should become part of your portfolio of evidence which will form part of your assignment submission.

Your SCRUM Team should meet at least weekly to review progress and when each sprint is completed to constantly review and plan the next phase of project development. You should produce minutes of all meetings and reproduce these as evidence. If you are working individually, then management documentation is still required, and you should reflect on how teamwork could benefit the exercise.

**Design, Development, Testing and Integration**

You should develop your system using a recognized developmental approach, such as ICONIX.

All design documentation should be kept as a portfolio of evidence, and will become part of your assignment as evidence to your claims within your report.

You should justify all design decisions you have made, and support them with appropriate theory and academic references.

You are encouraged to use prototypes to determine the most appropriate solution for a given set of criteria.

You will need to develop a test strategy, linked in with your SCRUM project management. The details of your testing, and all test results must be included within your evidence.

You are encouraged to use a configuration management/version control system for logging each version of software and its associated documentation that you produce. Evidence of your configuration management/version control system should be included in your report.

**Interim Goals**

By week 4 or 5 of the unit, you should, at the very least, be able to show the client representative that the system is feasible. In order to do this, you should be able to demonstrate, via a working prototype, that you can access, add to and otherwise manipulate appropriate data within a persistent storage medium of your choice.

By the same time, you should agree with the client representative initial user stories for implementation.

At various stages, your team will be provided with suggested user stories for implementation within the next sprint.

**Reflective Element**

Your reflective report should include your own individual evaluation of each of the stages of the development of your system. **Reflection and evaluation should form the basic structure of your report:**

**“This is what was done.**

**This was how it was done.**

**This was why it was done that way.”**

**Deliverables**

This will take the form of a reflective report but will be backed up by evidence of your work throughout the year. Its weighting is 60%.

This should address, as a minimum, activities of:

* requirements elicitation, analysis and expression,
* SCRUM project management,
* designing, developing, testing, integrating, refactoring etc.
* configuration management/version control

Where decisions have been made, they should be supported by theory and academic references, adding support and justification to your logic.

The evidence of work is used to build your report.

**Appendices of Evidence**

The claims you make in your presentation should be supported by well organised and referenced appendices of evidence of charts, diagrams, project management artefacts, tables, tests and their corresponding results. These are uploaded separately to Solent Online Learning.

**A working version of the system**

A .zip file (.tar and .rar are not acceptable) of all the source code and an executable file that will run on the University’s PCs.

You should also provide a link to a public git repository where your code is hosted.

It is recommended that you include a simple set of instructions or “How to run this program” section with your evidence. This will provide assessors with appropriate instructions on how to run your program from the executable you provide.

You should also demonstrate a web hosted version of the program which can be securely accessed.

**Design constraints**

The following non functional design constraints will apply to your design

* You should demonstrate a layered architecture with separation of concerns developed using Java web technology.
* Your design must include a common set of model classes and interfaces which are documented using UML. This API should be documented using Javadoc.
* The back end must be written using java technologies and deployable as a war file to a cloud hosted app server. You should consider using JSP, JSF, JAX-RS or equivalent for web presentation.
* You should use a suitable java persistence technology to implement a DAO layer (e.g. JDBC / JPA / JAX-RS etc.)
* You may use a client side framework to enhance the client experience (i.e. MVVC pattern with javascript and ajax) but the primary business logic must reside in the server side of the application.
* The app should be secure and demonstrate user login and roles.
* Demonstrate appropriate use of Error and Debug logging using a suitable framework such as LOG4j and / or SLF4J
* Demonstrate appropriate use of error handling and user input verification.
* Demonstrate automated testing and comprehensive test coverage of each layer
* Use a build system (maven to assemble and test you application.
* (stretch objective) use a continuous integration service (e.g https://circleci.com) to test you app as it is checked into git.
* Your system documentation should be hosted with the project, potentially in the form of markdown pages, Javadoc and suitable diagrams.

**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 (second or third attempt) 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.

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2o-assessment-policy-annex-1-assessment-regulations.pdf?t=1411116004479>

**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 a Student Achievement Officer for advice.

A summary of guidance notes for students is given below:

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-4/4p-extenuating-circumstances-procedures-for-students.pdf?t=1472716668952>

**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:

<http://portal.solent.ac.uk/support/official-documents/information-for-students/complaints-conduct/student-academic-misconduct.aspx>

**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:

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2s-university-ethics-policy.pdf>

**Anonymous Marking**

A copy of the University’s Policy on Anonymous Marking, process details and student guidance on submission sheet completion can be found on the following links, which are also uploaded on the Student Portal.

Fact Sheet: <http://portal.solent.ac.uk/documents/academic-services/policies-procedures-guidelines/anonymous-marking-fact-sheet.pdf>

Process: <http://portal.solent.ac.uk/documents/academic-services/policies-procedures-guidelines/anonymous-marking-process.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.

Policy: <http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2o-assessment-policy.pdf>

**Assessment criteria**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **D3 – D1** | **C3 – C1** | **B3 – B1** | **A4 – A1** |
| **Use of theoretical underpinning in design and project management**  **(LO K1, T1)** | Adequate understanding of the main concepts, theories and/or practice | Satisfactory understanding of the relevant concepts, theories and/or practice and their main implications and applications | Thorough understanding of abstract concepts, theories and/or cutting-edge practice and several of their implications and applications | Excellent understanding of abstract concepts, theories and/or cutting-edge practice; their implications and applications |
| Evidence of theoretical understanding of Agile development, TDD, ICONIX, GRASP Patterns, other patterns, management tools. | | | |
| **Developmental rationale & approach**  **(LO K1, P1)** | Few practical solutions sparsely argued/evidenced, mainly derivative and with little critical insight | Mostly relevant argument/evidence supports logical practical solutions showing some critical insight and limited creativity or originality | Practical solutions logically argued/evidenced, with some aspect of insight, creativity or originality | Insightful practical solutions closely evidenced showing originality and creativity in several aspects |
| Evidence of rationale for application of Agile methods, TDD, use cases, robustness & sequence diagrams, GoF and/or other patterns, management tools. | | | |
| **Development practice**  **(what you did and how professionally and competently done)**  (LO C1, P1) | Basic competence in all the required specialised practical and technical skills, and partial awareness of professional contexts and expectations | Achieves a basic level of competence in all the required specialised practical and technical skills, with more developed capability in at least one area, and some awareness of professional contexts and expectations | Competence in all the required specialised practical and technical skills, with indications of more developed ability in some areas and awareness of professional contexts and expectations | Consistent competence in all the required specialised practical and technical skills, with indications of mastery in some areas and clear understanding of professional contexts and expectations |
| Evidence of use of Agile solutions, automated testing and consideration of code quality, GoF and/or other patterns, management and control tools. | | | |
| **Team work**  **(LO P2)** | Limited evidence of practical understanding of team-working and leadership skills | Evidences good understanding of team-working and leadership skills | Evidences a high level of understanding of team-working and leadership skills and the consequences. | Evidences excellent understanding of all aspects of team-working and leadership skills |
| **Reflection and evaluation**  **(Your report)**  **(LO T2)** | Basic understanding of what was achieved. Some evidence of self-evaluation and reflection but with few critical insights | Good understanding of what was achieved. Satisfactory self-evaluation and reflection with some critical insights | Sustained reflection and self-evaluation generates a number of critical insights | Reflection and self-evaluation consistently critical and insightful throughout all phases of the development. |