# Southampton Solent University

# Coursework Assessment Brief

# Assessment Details

|  |  |
| --- | --- |
| Unit Title: | Object Oriented Design and Development |
| Unit Code: | COM504 |
| Unit Leader: | Dr Craig Gallen |
| Level: | 5 |
| Assessment Title: | Practical Assignment – Care Worker Safety |
| 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: | Week Commencing 23 September 2019 |
| Hand In Date: | COM504 10 January 2020  (16:00 GMT, 4 PM local time)  Demo Week beginning 13 September 2020 |
| Planned Feedback Date: | 4 Weeks after Hand in deadline |
| Mode of Submission: | on-line |
| Number of copies to be submitted: | 1 |
| Anonymous Marking | This assessment: is exempt from anonymous marking. |

# Assessment Task

**Case scenario (Care Worker Safety System)**

Current System

Personal Care Trust (PCT) has a mobile workforce of 'lone working' care assistants. Care Assistants each have their own vehicle and visit vulnerable elderly people in their homes to provide care assistance. Following several incidents where staff didn't turn up or were assaulted while doing their job, PCT have introduced a manual system to track their workers.

This system tracks the workers while they are travelling to their client and also while they are working at the client's home.

In order to ensure that their workers are safe, each worker is required to contact a central operations centre whenever they reach their client. After that they have to re-contact the operations centre every hour.

If the worker doesn't contact the operations centre every hour while they are at the client's home, the operations centre will try and phone them.

If the workers cannot be contacted, a second worker will be dispatched to make sure they are OK.

When the worker leaves their client, the worker contacts the operations centre and the safety of the worker is no longer tracked. However the worker is tracked for arrival at their next appointment.

Proposed system

PCT now want an automatic system which replaces this manual system. The system will automatically track workers on site and alarm if a worker has not checked in on time. The new system should allow variable tracking interval times to be set for each worker.

The new system will support manual operation where the remote worker simply phones in their status to an operator.

The new system will also support automatic operation which will provide each worker with an app or web page having three buttons. Each button will cause the app to send appropriate messages to the main operations centre in order to track the person on site.

* Button one 'Start Working with Client' must be pressed when the worker reaches site. The worker name, site address and GIS/GPS coordinates must be sent with the message. This starts a timer in the operations centre which will alert the operator if the worker doesn't press Button two every n minutes. This button will also start a timer in the app which will locally alert the worker with a beep to press Button two every n minutes.
* Button two 'I'm OK' must be pressed every n minutes while the worker is on site. This sends a message to the operations centre which resets the alarm timer.
* Button three ' I'm leaving Client Home' must be pressed when the worker leaves the site. This turns off the timer and informs the remote system that the worker has left site. The system will then track the worker to the next site

You must create both an operations centre service and a remote worker client.

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

1. **Central Operations Centre**

* The central operations centre should support MANUAL operation where the remote worker does not have an app but still phones in their status.
* The central operations centre should support a ReST service where the remote worker can use an app to update their status
* If a remote worker is unresponsive, the service should alert the operator to attempt to contact the remote worker.

1. **Remote worker Client**

* Implement the remote worker application. This can be a thick client app (e.g. Java Swing), an android app or a standalone web server which renders up the three buttons and allows the user to enter their name and current location.
* The app should alert the user to press the 'I'm OK' button after n minutes. It should use a ReST service to send notification messages to the main operations centre system.

**Design Process Requirements**

1. **Use Cases and Test plan**

You should document at least ten use cases covering each actor in your system. You should have use cases for manual operation and automatic operation for both the remote workers and the operations centre.

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 and Sequence Diagrams**

You should draw robustness and sequence diagrams to document how data is passed between the entities in your system

1. **Implementation**

* You may use a skeleton multi module maven project to give your project structure.
* You should use suitable Java persistence, Web and ReST frameworks for the backend server. Your back end artefact should be deployable on a web container as a war file.
* You should implement suitable a client which can interact with the backend service
* 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 interface classes with Javadoc

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 private project on Github (please make your lecturer gallenc a contributor). 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
* You should include instructions for a user to build, 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 sequence 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 represent 20% of the overall grade for this assessment and you will only be graded based on 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 | Reasonable attempt with significant inaccuracies and/or inconsistencies in your analysis and design.  Between 3-5 use cases presented 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 entering schedule also attempted  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 5 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 3 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. |
| **Unit 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 matching the use cases 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 (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>

**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>

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

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