

**DEPARTMENT OF COMPUTER SCIENCE**  
**COURSEWORK ASSESSMENT DESCRIPTION**

**MODULE DETAILS:**

Module Number:	08346	Semester:	2
Module Title:	Distributed Systems Programming		
Lecturer:	Dr P A Robinson		

**COURSEWORK DETAILS:**

Coursework Assessment Number:	1	of	1
Title of Assignment:	Factory system		
Format:	Demonstration	Report	3rd format
Method of Working:	Individual		
Workload Guidance:	Typically, you should expect to spend between	30	and 40 hours on this assessment
Length of Submission:	This assignment should be <b>no</b> more than:		<b>2000 words</b> (excluding diagrams, appendices, bibliography, code)

**PUBLICATION:**

Date of issue:	4 <sup>th</sup> March 2009
----------------	----------------------------

**SUBMISSION:**

ONE copy of this assignment should be handed in via:	White Box	If Other (please state method)	
Time and date for submission:	09:30	Wednesday 29 <sup>th</sup> April 2009	
If <b>multiple hand-ins</b> please provide details (as appropriate):	Demonstration times for software will be notified		

The assignment should be handed in **no later** than the time and date shown above, unless an extension has been authorised on a *Request for an Extension for an Assessment* (Mit Circs) form which is available from the Office or <http://www.student-admin.hull.ac.uk/downloads/Mitcircs.doc>. The extension form, once authorised by the lecturer concerned, should be sent to Amanda Millson.

**MARKING:**

Marking will be by:	Student Name
---------------------	--------------

**BEFORE** submission, each student must complete the **correct** departmental coursework cover sheet and attach it to your work, dependant upon whether the assignment is being marked by student number, student name, group number or group name. This is obtainable from the departmental student intranet at <http://intra.net.dcs.hull.ac.uk/sites/home/student/ACW%20Cover%20Sheets/Forms/AllItems.aspx>

#### ASSESSMENT:

The assignment is marked out of:	30	and is worth	30	% of the module marks
----------------------------------	----	--------------	----	-----------------------

#### ASSESSMENT STRATEGY AND LEARNING OUTCOMES:

The overall assessment strategy is designed to evaluate the student's achievement of the module learning outcomes, and is subdivided as follows:

LO	Learning Outcome	Method of Assessment <i>{e.g. report, demo}</i>
<b>1</b>	<i>Demonstrate research, selection and assessment of distributed system architectures</i>	Report
<b>3</b>	<i>Critically evaluate a range of contemporary distributed computing technologies.</i>	Report
<b>4</b>	<i>Specify, design and implement a distributed software application.</i>	Report/Demo
<b>5</b>	<i>Implement a distributed application using a contemporary managed environment.</i>	Demo

Assessment Criteria	Contributes to Learning Outcome	Mark
Appropriate analysis of problem and choice of solution technologies	1,3,4	15
Working software performing to specification	5	15

#### FEEDBACK

Feedback will be given via:	Verbal (via demonstration)	Feedback will be given via:	Select secondary method
Exemption (staff to explain why)			
Feedback will be provided no later than 20 working days after the submission date.			

This assessment is set in the context of the learning outcomes for the module and does not by itself constitute a definitive specification of the assessment. If you are in any doubt as to the relationship between what you have been asked to do and the module content you should take this matter up with the member of staff who set the assessment as soon as possible.

You are advised to read the **NOTES** regarding late penalties, over-length

assignments, unfair means and quality assurance in your student handbook, also available on the department's student intranet at:  
<http://intra.net.dcs.hull.ac.uk/sites/home/student/default.aspx>. In addition, **please note** that if one student gives their solution to another student who submits it as their own work, **BOTH** students are breaking the unfair means regulations, and will be investigated.

In case of any subsequent dispute, query, or appeal regarding your coursework, you are reminded that it is your responsibility, not the Department's, to produce the assignment in question.

# Assignment Details

## ACW1 – Factory system

### PROBLEM DESCRIPTION

You are to design and build a distributed application for Crunchibone Pet Foods Limited. The system will form an important part of the quality monitoring process for their dog biscuit production lines. The production line requires software at three different locations:

#### 1. *Weighing station*

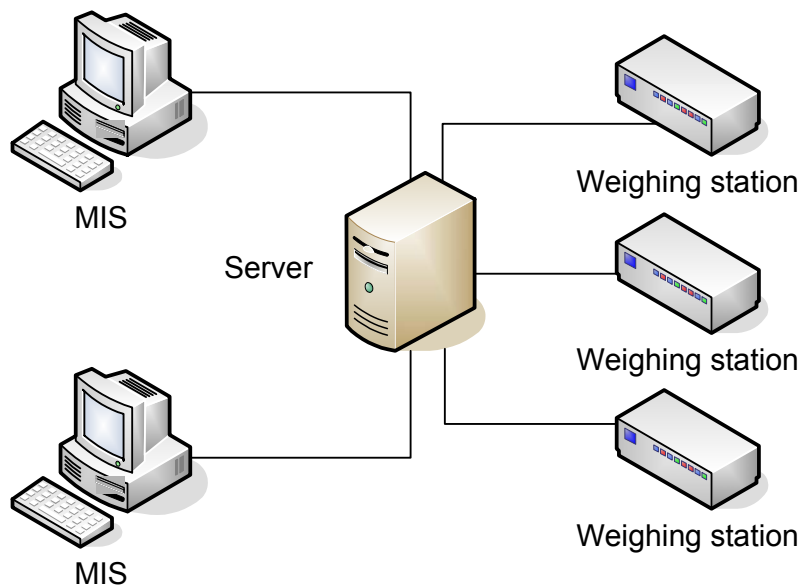
At a weighing station, filled bags of dog biscuits are presented to be weighed at approximately one second intervals. They are weighed automatically and the results are sent over the network to a server. There are many weighing stations in the factory. They are used for different products and different pack sizes at different times. At any given time, some may be operational while others are not. Weighing stations operate on Windows PCs and are connected to the company's LAN.

#### 2. *Data centre server*

The data centre server accepts data from weighing stations whenever they choose to send it, and sends summaries of the data to management information system terminals on request. The data centre server is connected to the company LAN and also to the internet.

#### 3. *Management information system*

The management Information systems (MIS) must present the data from the server in a readable form. The users want to be able to see simple reports of production volumes, percentage of underweight product bags, and so on. Several MIS stations may wish to look at the data at the same time. Some MIS stations may be operating on site and be connected by the factory LAN, while others might be at the company's head office, on a separate site. MIS stations may be connected or disconnected from the network at any time.



## **ANALYSIS OF THE PROBLEM**

There are several technologies available to you to develop this application. You have already built systems using some of them in the lab. It is up to you which technology or combination of technologies you decide to use, but you will be expected to justify your decision by thorough analysis of the problem you are trying to solve and investigation of the potential solutions. You will have to write this analysis and decision making process up in your report.

You will doubtless notice that the information available to you about the problem domain is far from complete. Where information is lacking, you should make sensible assumptions, and justify these in your report.

## **SOFTWARE YOU HAVE TO DESIGN AND BUILD**

You must write software for all three components of the system. All three must be Windows applications written using the .NET framework. Brief descriptions of these are given below, but it's part of your task to think about what is required and to come up with a sensible design. This will probably mean more functionality than is described here.

### *Weighing Station*

You must write software for the weighing station. Your software should allow a human user to set the nominal weight and name of the product being weighed at each station. This data must be passed to the server when it is changed. You will be provided with a library (downloadable from the module web site) from the manufacturer of the weighing equipment, which you must use in your weighing station code. This library will raise an event every time a bag of product is weighed. Your software must respond to these events and communicate the data to the server. When you pass data to the server, you should also include (at least) the date and time at which the measurement was taken and the nominal weight.

### *Server*

The server must receive data from as many weighing stations as wish to send it. It must store this data and be able to retrieve it on demand. It must provide data to the management information system terminals on request. The server will need to be multi-threaded. You are expected to write the server as a standalone application. You cannot assume that IIS will be running on the server platform.

### *Management information system terminals*

The MIS terminals must connect to the server and be able to request data to create reports. For the purposes of this project, only two reports will be required (though you may add others if you wish):

1. A report showing the last 10 readings from each weighing station.
2. A report showing the number of underweight bags for each weighing station in the last 24 hours.

Any MIS terminal should be able to request and present either of these reports at any time. The MIS application should be a standalone windows program, not a set of web pages.

### *A note on user interfaces*

Although parts of the system require interaction with the user, sophisticated user interfaces are not required, and will not attract extra marks. Credit will be given for good analysis, planning, design and implementation rather than eye candy.

### *A note on Ports*

In the Superlab, all the computers are firewalled, and will block incoming socket requests on virtually all ports. This should not affect you if all your applications are running on the same system, but means that you cannot easily test your application across the network. You may find that port 43 is open (it should be on most computers in the lab), in which case you may use that for testing between computers. You are advised to make your systems configurable at runtime, so that if required to do so you *can* demonstrate their use over multiple computers.

### *A note on third party libraries*

You may **not** use any third party libraries or code in this assignment except for the .NET framework.

## **DELIVERABLES**

You must supply the following deliverables.

1. A short report (no more than 2000 words) describing the design of your software. This must include details of the technologies you have used to create your distributed application, and your reasons for choosing them rather than others. One copy of this report should be submitted on paper.
2. A CD containing complete commented source code, executable files and any support files required to run your application. The CD should also contain a copy of your report, in MS Word or OpenOffice document format. Please submit your CD in a thin wallet or envelope rather than a solid 'jewel case'.

Please don't bother to bind your reports in fancy folders. A simple clear plastic wallet suitable for inserting in a ring binder is ideal. *Make sure both your report and your CD are clearly labelled with your name.*

## **ASSESSMENT**

You will be expected to demonstrate your application suite, showing the server, weighing stations and MIS clients operating at the same time. You will be notified in advance when and where the demonstrations will take place, and verbal feedback will be given during the demonstration.

In your applications, you will be expected to make use of the technologies you have been taught about in lectures: concurrency, threading and communication. You should be prepared to demonstrate these features.

**DEPARTMENT OF COMPUTER SCIENCE  
COURSEWORK FEEDBACK SHEET**

*(Students: submit this form with your coursework cover sheet and assignment.)*

Student Name or Number	<i>To be completed by student</i>
Module Number and Title	08346 Distributed Systems Programming <i>To be completed by staff</i>
Assignment Title	Factory system <i>To be completed by staff</i>
Mark Awarded	

**Extent to which your work meets the stated learning outcomes**

*To be completed by staff once work has been marked*

**Strengths and positive aspects of your coursework submission**

*To be completed by staff once work has been marked*

**Aspects of your work which could be improved**

*To be completed by staff once work has been marked*

Staff Signature\_\_\_\_\_

Date\_\_\_\_\_