

---

**Specification of Continuous Assessment Assignment (2008)**

**(30% of total module marks)**

**Change History of this specification:**
**14/02/2008: Initial issue**

1. Deadlines for submission of deliverables: .....	1
2. Deliverables .....	1
3. Teamwork .....	1
4. Problem statement .....	2
4.1 Overview .....	2
4.2 Requirements (sample from <a href="http://www.health-infosys-dir.com/dental_rfp.htm">http://www.health-infosys-dir.com/dental_rfp.htm</a> ) .....	2
5. *** Notes and guidelines *** .....	6
6. Specific tasks .....	6
6.1 Phase A Analysis, Design and System Test Definition .....	6
Task 1 ( <i>modelling</i> ) (11% of total module marks) .....	6
Task 2 ( <i>system test definition</i> ) (3% of total module marks) .....	7
6.2 Phase B Implementation and System Test Execution .....	7
Task 3 ( <i>coding</i> ) (11% of total module marks) .....	7
Task 4 ( <i>system test execution</i> ) (3% of total module mark) .....	8
6.3 Task 5 (project management) (2% of total module mark) .....	8

**1. Deadlines for submission of deliverables:**

Phase A: Tasks 1, 2, 5(a)      17.00 on March 20<sup>th</sup>, 2008

Phase B: Tasks 3, 4, 5(b)      17.00 on April 18<sup>th</sup>, 2008

**2. Deliverables**

Printed documentation for each constituent task plus, optionally, a demonstration at the end of Phase B. [A team may feel that a short demonstration (about 10-20 minutes long) would be useful in showing their system to best effect]. The documentation should be complete enough to allow assessment of the extent to which each specific task has been completed. (*There are 5 specific tasks – see below*).

**3. Teamwork**

Normally, the assignment should be done jointly by a team of 2, 3 or 4 people. Unless there are exceptional circumstances, the assignment should not be done by 1 person working alone as, in industrial practice, software development is almost always a group activity. *Please e-mail team membership to [ltuohey@computing.dcu.ie](mailto:ltuohey@computing.dcu.ie) by February 22<sup>nd</sup>, 2008.*

## 4. Problem statement<sup>1</sup>

### 4.1 Overview

A number of dentists work together co-operatively, and wish to acquire a software system for their group practice. They envisage that the system will support both their clinical work and management of the business. Thus, it will hold dental records for each of their patients enabling a dentist to review the dental history of a patient and to plan future interventions and procedures. On the other hand, it will maintain an inventory of the practice's equipment and material so that new purchases can be made in good time and out of date stock can be removed. It will also keep track of financial records, particularly in terms of patients' accounts etc. Thus, it is foreseen that the system will be used in different ways by clinical, secretarial and accounting staff. The system must be highly reliable and, especially, must guarantee safe management of dental records because of the potential impact on patients' health.

### 4.2 Requirements (sample from [http://www.health-infosys-dir.com/dental\\_rfp.htm](http://www.health-infosys-dir.com/dental_rfp.htm))

<b>Patient Database</b>
Support unlimited number of patients.
Provide ability to assign patient to multiple providers, clinics and locations.
Provide patient profile data fields for: multiple telephone numbers (e.g. cell phone, work phone, fax).
Provide ability to enter comments or specific instructions (e.g. nicknames or "do not call at work") for patient and/or family.
Provide ability to enter patient data as individuals or linked by family group, with responsible party identified.
Automatically update patient address or phone when one member of the family's address or phone changes (e.g. via family links).
Provide ability to search by: partial patient name.
More...

<sup>1</sup> Make sure to read following section "Notes and guidelines" to get extra clarification on what is required for the assignment.

<b>Scheduling</b>
Support manual or automatic computer searches for appointment time needed by provider, considering the availability of the provider and the facility.
Provide ability to schedule a family (e.g. two or more patients) with one entry instead of entering each appointment individually.
Support appointment searching for specific requests such as Saturday morning only.
Automatically alert user if an attempt is made to schedule a recall in advance of 3rd party allowances.
Generate a list of instrument cassette types needed, by provider, as interpreted from the scheduling module.
Support multiple types of recalls (e.g., three month period check, six month denture check).
Support reactivation tracking and reporting for inactive patients (e.g. list of patients not seen by the office for 18 months).
Print time utilization reports which compare planned time vs. actual.
Track date lab work is sent/due/returned.
More...

<b>Treatment Plans</b>
Support on-screen treatment planning.
Provide ability to generate multiple treatment plans for one patient.
Provide ability to retain deleted and/or changed treatment plan items, and identify them as deleted and/or changed.
Provide ability to store explanatory notes in association with individual treatment plan items.
Provide ability to sort and re-order treatment plans as needed.
Provide ability to print incomplete treatment plans by: provider, procedure or user defined criteria.
Support electronic transmittal of treatment plans to third party carriers for pre-determinations.
Automatically calculate insurance benefits and patient financial liability when

<b>Treatment Plans</b>
entering procedures on a treatment plan.
More...

<b>Clinical Records</b>
Allow the practice to customize the health history questions.
Support method to record: soft tissue and hard tissue pathology.
Support sharing of family health history records among family member records.
Provide ability to note specific allergic reactions (e.g. anaphylactic shock, hives or upset stomach).
Provide ability to calculate DMFT and DMSF from odontogram.
Support recording all ADA approved Treatment and Diagnostic codes with associated modifiers particular to the specialty being charted.
Provide historical comparisons of clinical observations (e.g., changes in periodontal data over a two year period).
Maintain audit trail of patient information disclosures, by whom, why, and when.
Maintain a computerized sedation log.
Provide ability to store digital diagnostic images for viewing on workstations.
Provide ability to store altered images with time/date stamp and markings denoting them as altered images.
Print post-sedation instructions.
Provide ability to scan and store documents (e.g. signed consent forms, letters, medical consults).
More...

<b>Patient Billing</b>
Provide ability to enter and maintain patient demographic and financial data for billing purposes.
Display the following information upon patient "check-out": next appointment, recall information, tooth number and surfaces.
Provide ability to enter an alternate fee by overriding default fee.

<b>Patient Billing</b>
Support entry of "explosion codes" where one designated office code embodies all appropriate CDT codes and procedure descriptions.
Support multiple billing cycles for batch statement printing (e.g. aging date, alphanumeric by name, etc.).
Provide ability to print unique statement messages on patient statements.
Support user-defined filtering and sorting of financial reports.
Print customized collection letters or dunning statements by patient or group of patients meeting user-defined criteria (e.g. balances over 120 days).
Support financial promise tickler file.
Support Electronic Funds Transfer (EFT).
Support HMO / PPO / Capitation billing and processing.
More...

<b>Practice Management</b>
Print daily patient visit report listing patients seen per day, production and collections.
Print daily procedures by producer detailed report with procedures performed listed and counted for the day and with month-to-date totals.
Print management report showing collection / production ratio
Print average patient case fee report (e.g. average production and collection per patient by provider).
Print referral report showing patients referred, treatment planned, treatment performed, fees and collections by referral source.
Support sub-practice reporting with user-defined breakdown levels (e.g. all funds are directed to one deposit account, but production and collection reported by sub-practice).
Track referrals of patients sent out of the office and whether referred specialty care was followed through.
More...

## 5. \*\*\* Notes and guidelines \*\*\*

(i) Clearly, for the purpose of the assignment, it would *not* be realistic to build a system that meets all the requirements listed in section 4. **Hence, a very important step in Task 1 will be for each team to DECIDE on the quite limited subset of the requirements it will implement.** It is possible that a team may decide to include some requirements that are not listed in section 4 or to re-phrase given requirements. Such decisions and the reasons for them should be recorded briefly in the project log (see **Task 5** below).

(ii) In an actual project, it would be necessary to have a process of consultation with the customer and with the intended users of the system (dentists, secretaries etc). The purpose of this consultation would be to establish a clear and detailed requirements specification as well as major constraints to be satisfied by the system. *However, this is obviously not practical for the purpose of the assignment. Instead, you have discretion to decide on system details as well as on the overall scope of the system.*

## 6. Specific tasks

### 6.1 Phase A Analysis, Design and System Test Definition

#### Task 1 (*modelling*) (11% of total module marks)

Develop a set of UML models and diagrams based on the above problem statement.

This should include,

a) A use case model consisting of a use case diagram, depicting actors and use cases, with a corresponding description for each use case depicted.

*Note: A completely comprehensive list of uses cases is not expected. As a guideline, about 8 use cases corresponding to the requirements subset that a team has decided on would be about right, depending on the size of each use case.*

b) A class model – it is expected that candidate classes will be found using the noun identification technique and that some high level “real-world” associations will be recognized. Then, the CRC technique (or equivalent) should be applied to refine the model by identifying responsibilities and collaborators for each class.

c) Interaction diagrams to show, in particular, how the use cases are realized through interactions of objects of the various classes.

d) State diagrams to depict any significant states, and corresponding transitions, that have been identified for classes.

e) Activity diagram(s) to show, for example, aspects of the overall flow of activities within which PMS functions.

f) Pre- and Post-conditions and/or class invariants: While not part of UML as such (but can be depicted within it), it would be very useful to identify any significant such conditions or invariants.

g) Deployment diagrams to depict how, and on what machines, the completed software will be deployed.

**Note 1.1:** *For the purposes of the assignment, it is sufficient that the software be executable on a standalone machine although in reality it might be deployed over a distributed system.*

**Note 1.2:** It is expected that a suitable UML software tool (e.g. **WithClass**) will be used in this task.

**Note 1.3:** It is to be expected that there will be iteration when performing the above, especially b), c) and d). Also, when performing Task 3 (coding) you can expect to make adjustments of the UML model.

### **Task 2 (system test definition) (3% of total module marks)**

Define system test cases of the software based on exercising the various scenarios within each of your use cases (see Task 1).

## **6.2 Phase B Implementation and System Test Execution**

### **Task 3 (coding) (11% of total module marks)**

Implement the UML models and diagrams of Task 1 in Java, resulting in source code for each class and corresponding executable code.

**Note 3.1:** It is not necessary (but of course not forbidden!) to use a database system to store any persistent data that are needed (such as sample lists of patient details). It is quite acceptable to store such data in simple files; e.g. in this way one could instantiate objects as part of initialization.

**Note 3.2:** In a real system, the user interfaces for the different users of the system would likely be quite separate. However, for the purposes of the assignment, it is acceptable to have a relatively simple user interface.

**Task 4 (system test execution) (3% of total module mark)**

Execute the system test cases you defined in Task 2 (including defining actual test data if this has not already been done in Task 2), and document the test results.

**6.3 Task 5 (project management) (2% of total module mark)**

Each team should keep a succinct 'management' record or log of its weekly activities. This should include *brief* entries about sub-tasks started and completed, job responsibilities within the team, decisions made, actual versus planned progress, risks, lessons learned and so on. The corresponding part of the record or log should be submitted as part of the deliverables for Phases A and B.