|  |
| --- |
| **INB201 Scalable Systems Development** |
| ***Project – Integrated Hospital Information System***  **Worth:** 50%  **Due:** Week 14 (Exam Period), Semester 1 2014  **This project is to be completed as a team.** |

**2 March 2014**

**Synopsis:**

This project will require your team to create a complex information system using the Scrum systems development methodology. The development of this system is a semester-long team-based activity that culminates in the submission of a working product and associated documentation.

**Restrictions:**

This project is to be completed as a team effort. Teams are to consist of 5 students. (Where it is unavoidable, staff-approved teams of 4 or 6 students are permitted.) Students must use the Scrum systems development methodology. However teams may use any available software tools and programming languages to complete the project – see later.

**Requirements:**

The state of Queensland is to build a new Children’s hospital in Townsville. However due to recent incidents in existing hospitals where patient test results have gone missing and paper records have been lost, the hospital is to employ a state of the art computer and communications network throughout all of its processes. Essentially the new hospital is to be paperless.

Your task is to design and implement an Integrated Hospital Information System. This system will be a comprehensive, integrated information system, designed to manage the administrative, financial, medical and clinical aspects of the hospital. It is not for use by the general public. Your system must be functional and demonstrable to the hospital stakeholders at the end of the semester.

The following are the essential **functional requirements** of the system:

* The system must track patients from admission to discharge. The system should consolidate all patient information including patient histories, test results, observations, doctor’s orders, surgeries, etc.
* The system should be able to track financial information for patients including calculating and claiming the Medicare rebate or refunds from private health insurance companies. This means patients are only charged for the “gap”.
* The system must have secure *access control*. Staff should be able to authenticate to access the system (using a password-based system) and be given privileges depending on their role, i.e. a *role-based* access control approach should be developed.
  + Doctors’ notes and orders should be able to be entered.
  + Nurses’ observations should be able to be entered.
  + Test results including x-ray scans (students can use jpg files to test this functionality) from medical technicians should be able to be entered and viewed by authorised persons.
  + Receptionists should be able to enter other patient personal information like addresses and payment information.
  + System Administrators should be able to configure and query the system without having to change the code.
* The system should be used to schedule patient activities and hospital resources. This includes use of patient beds, operating theatres, MRI and x-ray machines etc., so that for example two surgeries are not scheduled in the same theatre at the same time or 10 patients kept in a 2 patient room.
* Often patients may need to be transferred to another hospital. Relevant data should be able to be exported to PDF files.
* Hospital administrators need to be able to configure the system, query the system for specific information and to generate reports regarding the data in the system.

**Note:**

1. The listed requirements cover a range of functionality. Depending on your team’s capacity, you might not be able to complete all the requirements. Therefore, you need to discuss with your customer to prioritize the tasks.
2. The requirements can change over time. New requirements can be added if the customer sees the need.
3. You are encouraged to discuss with your customer about including extra functionality to the system. You will receive marks accordingly for the agreed-upon extra functionalities.
4. You are discouraged from adding whatever functionalities simply because you like them. You have to discuss with your customer first, any new functionalities you plan to add to the system.

**Your tutor will act as your customer in this software development project.**

This project will require a multi-disciplinary team. You will need business analysis and business process knowledge, interaction design skills, programming skills, database skills, and computer security knowledge. It is possible that you do not currently have all the knowledge required to produce the specified system. You are encouraged to learn new skills to complete this project.

Because you are to work as a team in a professional manner, there are these additional requirements of the project.

***Non-functional requirements***

The following are the **non-functional requirements** of your project:

* You must ensure that all user interfaces are suitable for non-technical users.
* You must ensure that all code is written and formatted according to a source code standard that your team chooses.
* You should ensure that the system is designed in a proper manner to enable reliability, availability and scalability.

***Choice of Platform***

*Hardware:* Normal PC (desktop or laptop)

*Software:*Your team will need to choose the bundle of software you will use: Operating System, Web Server (where applicable), DBMS, and Programming Language. While developing a web application is preferred in terms of a *real-world* application, we recognise that not all INB201 students have studied web development. Therefore, the alternative of developing a thick client (desktop) application is permitted, provided that this is written in C#. (By itself, the choice between a thin client and a thick client solution will not directly affect your marks – see the Marking Criteria document, once it has been released.)

It is expected that most teams will use one of:

1. The WISA (Microsoft) bundle: Windows, IIS ([Internet Information Services](http://en.wikipedia.org/wiki/Internet_Information_Services)), SQL Server, and ASP.NET.
2. The LAMP bundle: Linux, Apache, MySQL, PHP or Python.
3. The WAMP bundle: same as LAMP but using Windows instead of Linux.
4. A WinForms application: Windows, SQL Server, MySQL or Oracle, C#. If you create a WinForms application, then Microsoft Access is ***not allowed*** as the DBMS, because it has very limited security features. Instead, you must use a more “heavy duty” DBMS, such as SQL Server, MySQL, or Oracle.

If your team wants to, it is permitted to use a content management framework (CMS), such as Drupal or Joomla. This is generally *not advised* if no-one on your team has prior experience with such a CMS, because of the extra learning curve. Also be warned that while a CMS can make it easier to get started, it can also make it harder for you to fully satisfy the project requirements. To do that, it is likely that you will have to modify the CMS’s program code. Making sense of other people’s code, so that you can change it, is likely to be much harder than developing your own program code.

**What to Submit**

As well as giving a final, oral presentation at the end of semester (20 minutes maximum), your team will be required to submit:

1. Product Documentation, in hardcopy, as well as a softcopy in MS Word or PDF format.
2. Source and executable code, with installation and configuration information.
3. Demo of a working system (a video or a series of screen shots).

The final system documentation is partly aimed at informing the users of the system about how they can interact with the system. It should also include the requirements and design of the system, testing plan and results, and discussion on the quality assurance (e.g. code review). There should also be information regarding the installation and configuration of the system. You can use visual aids such as screenshots in the User Guides. You must submit source code and executable files. Clean your code and documents before you submit them to make sure that only useful documents are included. (More information describing the required, overall structure of this documentation will be released early in the semester.)

The product you submit will be assessed on how well you have designed and implemented the system **according to the requirements agreed by your team and your customer** (your tutor/customer has the rights to add or remove to/from the requirements listed above and re-prioritise them). Staff will base their marks on observations and questions during the final presentation, and the submitted items.

**How to Submit**

The final presentation is to be conducted by your team during week 14, at a time to be specified by the Unit Coordinator. It is expected that **ALL** students in your team will attend the presentation.

The hardcopy of the Product Documentation should be submitted as a professional-looking document, in some kind of binder. The softcopy of the Product Documentation, source and executable code, and demonstration of the working system are to be submitted on a CD/DVD ROM – or on a USB device. Your team are to submit both the hardcopy and the CD/DVD ROM or USB device via Assignment Minder by the end of week 14. (If you submit a USB device, make sure that it is securely attached so that it cannot fall out of the document wallet that you hand-in at the Assignment Minder desk.)