Phase 2 Integrated Project

Course Title	Phase 2 Integrated Project
Project Code	PP2E v1-0 Project 2005-0430
Project Title	Northern Lights Hospital
Pages	11 pages (plus cover sheet)
Released	April 30, 2005

Phase 2 Integrated Project

Project: Northern Lights Hospital

INTRODUCTION

You are a junior Software Developer working for a small software solutions firm, CDISoft. In the first few months, the firm gave you several small prototype projects to assess your abilities. Your work has been outstanding, and CDISoft is now ready to entrust an application to your care.

NorthernLights Hospital is a small health care facility located in northern Canada. Due to lack of money, they have had to make do with outdated paper and DOS-like programs to track patients. Although the employees are very familiar with the old system, and rarely make mistakes (or know how to fix them if they do), NLH feels its time to update the system. The hospital board was very impressed with the proposal to design, develop and maintain the patient tracking system put forward by the senior developer and awarded the contract to CDISoft.

When putting together the proposal, the senior developer was mindful that a patient tracking system is only the first step in a future major scale application. Normally, a smaller scale application such as that required by NLH would most likely be 2-tier using an Access database. However, if designed as an n-tier solution using SQL Server for scalability, it would be possible to easily modify and expand it with minimal disruption to staff plus the application could be sold and modified to suit other health care facilities. NLH is agreeable to such an arrangement because CDISoft intends to absorb the additional investment costs with a view to recouping them later.

Initially, Kent Tellme, a coworker, was assigned to work with the senior developer. Unfortunately, just before starting development, the senior developer suddenly took ill due to a stroke and is unable to work for the next several months. Kent Tellme was promoted to intermediate developer responsible the NLH project. Within days, Kent suddenly left CDISoft for another job. That meant the development would need to be entrusted to a contract worker or to one of CDISoft's three junior developers. CDISoft has chosen you to complete what Kent started while they search for replacements for the senior and intermediate developers. Of course, you want a promotion to intermediate developer, so you are willing to do whatever it takes to develop the business solution that NLH and its employees require. You are aware you will need to quickly learn techniques that you may not be familiar with yet.

YOUR ROLE

CDISoft wants you to use whatever Kent Tellme has already developed. Kent always had a bad habit of not documenting his work so you will need to carefully review it. You need to familiarize yourself with the business rules, decipher and apply what Kent has completed, and finish building the final release of the patient admission system for NLH.

OBJECTIVES

The objectives of this project are:

- Design, adapt and document an n-tier application
- Design, adapt and build a SQL Server database
- Design, adapt and build a user-centric interface
- Build an n-tier application
- Debug and handle errors to produce an error-free application

TIME REQUIRED

You will require 30 hours to complete this project.

However, to complete this project on time, you are expected to work beyond regular college hours. A good estimate is two to three hours of homework per school day, which is 12 to 18 hours of homework over the next six days.

MATERIALS REQUIRED

To complete this project, you require:

Hardware

- One PC per student with an Internet connection and access to a printer
- Pentium II 450 or higher processor (Pentium III 600 recommended)
- Windows 2000, XP Professional or NT4 (with SP 6) operating system
- Minimum 64 MB RAM for NT4 Workstation; 96MB RAM for 2000 Pro; 160 for NT4 Server; 160 MB for XP Pro; 192 MB RAM for 2000 Server;
- Minimum 550MB hard disk space for the operating system and applications
- CD-ROM
- SVGA monitor
- 1 blank diskette

Software

- A Windows operating system (Windows 9x, Windows 2000 Professional or Windows XP) installed on the computer
- Microsoft Office Professional Edition installed on the computer
- Microsoft Visual Studio .NET installed on the computer
- Microsoft SQL Server installed on the computer or network
- AVG Anti-virus (or similar antivirus software)
- Microsoft Internet Explorer version 5 or later

Textbook

- *Systems Analysis and Design in a Changing World*, Satzinger, Jackson, Burd
- *SQL Server 2000 Database Design and Implementation*, Thomas Moore
- *Beginning Visual C#*, Karli Watson

BUSINESS REQUIREMENTS SPECIFICATION

General Description

The NorthernLights Hospital (NLH) tracks hospital admissions. The system keeps track of each patient, each patient's doctor, next of kin, and private insurance company, if any. The system also tracks the bed and ward to which the patient is assigned, as well as any additional costs incurred by the patient such as a TV or private room. Typically, during development, the n-tier application runs on a single computer. When the application is released, the tiers are installed separately: the user interface objects will run on desktop computers, and the business objects and database will run on one or more servers.

The NorthernLights Hospital system uses a simple login system. For security:

- Only the hospital administrator can add or delete doctors, and produce invoice reports to bill patients for extra amenities.
- Only admissions can add admissions records. This involves adding a new patient, or finding a previous patient, assigning a doctor and a bed in an appropriate ward, recording the next of kin and private insurance company, and indicating what additional amenities such as a TV that the patient has selected.
- Only doctors can discharge patients.
- Only nurses can list patients by ward.

Data Requirements

The NLH application requires an Employee Login and the NorthernLights Hospital System.

During discussions with the NorthernLights Hospital board, Kent Tellme identified possible data required and produced an Access database as a development model.

Business Rules

NLH wants to observe the following rules:

- Beds are identified by a ward prefix and a number, for example S01 for Surgery, bed 1.
- Bed types include private, semi-private and ward. Ward is usually chosen.
- NLH cannot admit more patients than the total number of available beds in all wards.
- Patients are never deleted. When the doctor discharges a patient, the bed to which the patient was assigned is made available.
- Admission records are never deleted.
- If a patient has no private insurance coverage: If there is no ward-type bed available in any ward, admissions can assign the patient to a semi-private bed in the preferred ward, if available, at no additional cost. If all semi-private

beds are occupied, admissions can assign the patient to a private bed in the preferred ward, if available, at no additional cost.

- If a patient has no private insurance coverage, the patient will pay for a semi-private or private room if a ward-type bed is available in any ward but the patient elects a semi-private or private room instead.
- Surgical patients are automatically admitted to Surgery ward if a bed of the desired type is available. If a bed is not available, the user can select another bed type, or another ward, if available.
- Patients 16 years of age and under, if they are not scheduled for surgery, are automatically admitted to Pediatrics ward if a bed of the desired type is available. If a bed is not available, the user can select another bed type, or another ward, if available.
- Daily rates: Semi-private $267.00, private $571.00, TV $42.50, Phone $7.50.
- Nurses require a patients list by ward. The report should include the patient ID, name and, at a minimum, the doctor ID. The nurse should be able to select one or all Wards for the report (Use Crystal Reports).
- The billing report is generated by the administrator. The report should include the admission ID, patient ID, name, and number of days in the hospital. It should also include and identify any additional extra charges. This report requires a total.

Program Design Requirements

You are responsible for:

1. Providing modeling documents for the system.
2. Updating, normalizing or de-normalizing the Access database and converting it to a SQL Server database. Creating two stored procedures.
3. Designing any outstanding forms for the system. Update the existing forms to include any necessary menus, status bars, etc.
4. Reviewing the code written by Kent Tellme. Debug, correct and comment code where required. Reuse code wherever applicable.
5. Writing and commenting new C# code where required, and using ADO.NET
6. Designing and generating reports.

Your Task

You are responsible for completing the solution's development on time and on budget. You must work quickly. Valuable time has been lost due to the loss of the senior and intermediate developers. Review the work Kent Tellme has produced and compare it to the requirements outlined above.

Database Design

About 1 hour should be enough to complete this step.

Study the requirements, the database and the relationships between tables. Determine if one or two databases and if two servers are required, keeping in mind the limited resources of NLH and the additional cost to CDISoft. Identify any change which is required before transforming the database to SQL Server (for example, any de-normalizing required).

- Write a half-page report.
    1. State your reason(s) for using one or two databases or servers, and note the advantages and disadvantages of your choice. If you decided to use one database, identify any savings or future costs. If you decided to use two databases, identify what each server will house. Justify the cost of the architecture.
    2. State your reason(s) for changing, or for not changing, the structure or data types in the database. Identify structural database changes in a revised relationship diagram (additional report page). Note any data type changes.
    3. State where you will manage most business rules and why – in the database or in the business objects (NLHospitalLibrary).

System Design

About 3 hours should be enough to complete this step.

1. Review the system requirements and the database. Determine which parts of the system Kent has addressed and which parts are outstanding. This will help you determine what you need to develop from scratch
    - For Admit Patient only create:
        1. An activity diagram
        2. A three-layer sequence diagram
    - For the entire system, create a package design.

User Interface Design

About 2 hours should be enough to complete this step.

Study the user interface components created by Kent to determine how you can incorporate them into a user-centric interface design of your own.

- Sketch a draft storyboard
- List and design Windows form(s) required to Admit Patient. NOTE: You are not required to produce interfaces for Extra Rates, NextOfKin or InsuranceCompanies tables if you decided to use them. You can populate the tables through Server Explorer.
- List any global properties such as backcolor and roughly sketch Windows form designs.

Kent Tellme created a parent Windows form, NLHBase, from which other Windows can inherit (instead of from `System.Windows.Forms.Form`). You set any form property or add any control in the parent form and through inheritance it will be applied to the child form. For example, if you want several forms to use the same Quit button, you can add the Quit button to the parent form. Then you include code to inherit the parent form in each Windows form that you want to use the same control. Note that you cannot edit or delete an inherited control, but you can change property settings in the child Windows form.

You will use Kent's NLHBase form for all global properties and controls.

KEEP THE APPLICATION SIMPLE: Be careful to keep the complexity low so that you can complete the interface in time.

Programming Logic Design

About 4 hours should be enough to complete this step.

Study the programming code written by Kent to determine what sections you could reuse in to other parts of the application. This development technique is typically used in industry to save time and money, and requires you to use your creative problem solving abilities.

Unzip the Visual Studio .NET solution to a location identified by your instructor (if another student is using the same computer at a different time, you need to make sure your files are not accessible to that student). The solution includes three projects: NLHBaseWindow, NLHospitalLibrary, and NLHospital. Print out the existing code. You can expect that most of Kent's logic is reasonably sound however there may be some minor errors. You will reuse (copy, paste, edit) applicable code.

- List the form(s) you plan to create to Admit Patient. Across from the new form(s), identify an existing form and class whose code you can reuse.
- Include flow charts and/or pseudocode to describe decision making required for Admit Patient. For example, you need to document how the system will look for an available bed.

Update and Transform the Database(s) to SQL Server

About 2 hours should be enough to complete this step.

Based on your planning, make any changes to the Access database and transform the database(s) to SQL Server. Write a stored procedure to use with the Patient listing.

- Print out the stored procedure created in SQL Server.

Update and Create the Windows User Interface

About 3 hours should be enough to complete this step.

Based on your storyboard and interface design documents, modify or create Windows forms complete with any menus, toolbars, and controls. Add navigation code to appropriate buttons and menu choices. Use the NLHBase form and inheritance to manage universal properties.

Keep track of the time you are spending in this step. If required, leave out any controls or features you indicated in the design sketch that are not critical to the functioning of the application (for example, Status bars). It is CDISoft's policy that after successful implementation, you can always incorporate these 'value-added' features at additional cost to the client. In the case of NLH, which is not paying the full cost of the application, the cost of any additional features would be offset by future sales to other health institutions.

Revisit your design documents and note any additional revisions you made. In the interface design sketches, note any non-critical feature not incorporated as 'future' features to avoid losing marks.

Debug and Write Code for the Application

About 15 hours should be enough to complete this step.

Before you reuse any existing code, or add any new code, you need to review Kent's code and update or fix any obvious errors. Additionally, you must add comments. Try to phrase your comments so that you can copy and reuse them elsewhere in the project.

Build and debug the application. If there is no form containing an Add record method for a table (for example, NextOfKin), add data directly to the tables through the Server Explorer in Visual Studio .NET.

After you are satisfied that the application written by Kent runs correctly, copy, paste and edit any code you identified that is suitable for reuse.

Finally, add any new code required. Make sure you test and add error-handling as you work.

Use Crystal Reports and the stored procedure for the patient listing and any method, for example a form with print button, for the patient Invoice. Print out each report.

If you have trouble meeting a deadline, a partially built application that works very well is much more acceptable to the client than an application that looks pretty but doesn't do anything.

Revisit your design documents and note any additional revisions you discovered you need to make to meet the deadline.

HINTS AND TIPS

1. The DateTime data type does not automatically call its constructor as do the more commonly used data type variables such as string and integer. To use a new DateTime, you need to create it as you would any other object. Read through Kent Tellme's code to learn how to use to the DateTime object.
2. To create a parent form, you need to:
    - Add a new project to the solution.
    - Design and create the parent Windows form.
    - Select the new project name in Solution Explorer.
    - From the **Project** menu, select **Properties**. Under **Common Properties**, **General**, locate, in the **Applications** section, **Output Type**.
    - Change **Output Type** to **Class Library**.
    - Build the project to create a **.dll** file.
    - Select the Windows interface project for example, NLHospital, and add a reference to the new Windows form .dll file.

IF YOU HAVE TIME

You can add to your application's functionality. Make sure you keep a copy of your completed thus far before you attempt any additional work. Add first-draft design documentation where applicable.

2. Add user input validation in the business objects.
3. Add an audit table to track changes made to admissions records. The audit table should include the username of the person who made the changes.
4. Identify and incorporate any triggers in the database. Print out the triggers.
5. Identify and create additional stored procedures. Print out the stored procedures.
6. Add a Surgery list report. Only doctors and nurses can produce a list of patients due for surgery. The nurse or doctor should be able to generate a report for any specific date. The report should contain the patient ID, name, surgery date and doctor ID.
7. Create a second Windows parent form from which the Crystal Reports Windows forms may inherit.
8. Allow the Administrator to update the Extras Rates by adding a user interface and appropriate code.
9. Allow Admissions to update the Insurance Companies by adding a user interface and appropriate code.
10. Add a feature to record the type of diet patients are on: standard, restricted salt, low-fat, restricted sugar, Kosher, liquid.
11. Add a method(s) that will automatically generate the next AdmissionID, and/or next DoctorID.
12. Any feature of your choice that NLH would consider useful in the application as approved by your instructor.

MARKING SCHEME

You are graded on the following components:

Project component	Points
Design	**30**
Database Design: database changes, server architecture, business logic placement	**3**
System Design:	**10**
• Activity diagram	4
• sequence diagram	4
• package diagram	2
User Interface Design:	**7**
• Storyboard	5
• list of new Windows forms, and of global properties and controls	2
Programming Logic Design:	**10**
• list of new forms identifying existing code for reuse	2
• flowcharts and pseudocode	8
Database Implementation	**5**
All changes, if any, made per design documentation.	1
Microsoft Access database(s) correctly imported into SQL Server.	2
Stored procedure correct and printed.	2
User Interface Implementation	**10**
Windows forms and controls meet specifications noted in design requirements submitted.	5
Functionality meet specifications noted in design requirements	5
Programming Application	**45**
Identification, correction and debugging of existing code	8
Application adheres to design documentation submitted	8
Identification and proper reuse of existing code	6
Correct coding techniques	10
Use of appropriate error handling	5
Appropriate internal documentation (comments)	4
Invoice and Crystal Report out meets specifications, accuracy of data represented, layout	4
Conclusion	**3**
Presentation Quality (presence of all required elements including title page, neatness)	**2**
TOTAL	**100**

WHAT TO SUBMIT

Your project must contain:

- Title Page
- Project Description
- Written database report
- System design diagrams documentation:
    - An activity diagram
    - A three-layer sequence diagram
    - A system package design.
- User interface design documentation:
    - Draft storyboard
    - List of additional Windows forms required.
    - List of global properties and controls to be applied in the parent NLHBase Windows form.
- Program logic design documentation:
    - List of forms with reusable code identified for use in new forms.
    - Flow charts and/or pseudocode
- Print-out of the stored procedure.
- Print-out of the reports.
- Print-out of program code written in C#
- Diskette containing the solution files and final Access database required for the NorthernLights Hospital system (zip or compress the files).
- A list detailing clearly what user input is validated (if any), and what input is not. This helps your instructor determine what type of input to enter to run and evaluate your project properly.
- Conclusion

PENALTIES

- For each day that a project is late, 5% will be deducted.
- Projects that are more than three days late will earn a maximum score of 60%.
- Projects that contain a virus must be resubmitted and will earn a maximum of 60%.