Course Title	Processing Information
Project Code	PRIE v1-1 Project 1
Project Title	MultiLease Management System
Pages	16 (plus the cover)
Released	
Revised	11/03/2015

Processing Information

MultiLease Management System

INTRODUCTION

You are an Information Technologist working for a small software solutions firm. You have been assigned to help develop a customer vehicle leasing management system.

MultiLease is a small vehicle leasing company that specializes in one particular make of vehicle. The owner, Mike Lee, is considering expanding his inventory to include other manufacturers. Before committing to any new arrangements, Mike wants to improve the sales and account management system. Currently, each department completes its own paperwork and maintains its own records with its own software application. Departments communicate details by telephone or paper. Several costly mistakes have resulted in the loss of too many good customers.

To improve customer service, encourage repeat business and avoid communication errors, MultiLease' owner wants one application that provides employees with access to all administrative functions. The new application will provide an integrated interface, so

Sales Associates can record new vehicle lease contracts, and the Account Managers can easily obtain customer account information. Mike wants access to all features of the new application. Naturally, built-in security features prevent unauthorized individuals from obtaining sensitive information. Security issues are described in greater detail later.

Mr. Lee's long term plans include company expansion into new markets. He would like to open new branches in the future.

You are part of a team of IT professionals hired to implement a business solution for MultiLease's administrative problem. Your team understands the importance of integrating current business practices, rules and documentation into any new system.

OBJECTIVES

During this project, you will:

- Interpret and update modelling and design documents to build an application
- Create user-centred Windows forms
- Write programming code based on structured and object-oriented concepts
- Access and manipulate a SQL Server database through an application
- Maintain assembly information
- Debug and handle errors to produce an error-free application

TIME REQUIRED

You are given 20 hours of class time for 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 8 to 15 hours of homework over the next four days.

MATERIALS REQUIRED

The following software and textbook resources are required:

Software

- Un système d'exploitation Windows 7 installé sur l'ordinateur.
- Microsoft Office Édition Professionnel installé sur l'ordinateur
- A Windows operating system (Windows 7, Windows 8) installed on the computer
- Microsoft Office Professional Edition installed on the computer
- Microsoft Visual Studio .NET.
- Microsoft SQL Server 2008

Textbook

- *Beginning Visual C#*, Karli Watson.

BUSINESS REQUIREMENTS SECTION

General Requirements

The MultiLease Management System enables Sales Associates and Account Managers to effectively manage customer leases. The systems analyst on your team has drafted a system chart that describes the proposed application based on interviews to discover Mr. Lee's requirements, company documents, and employee work habits and requirements.

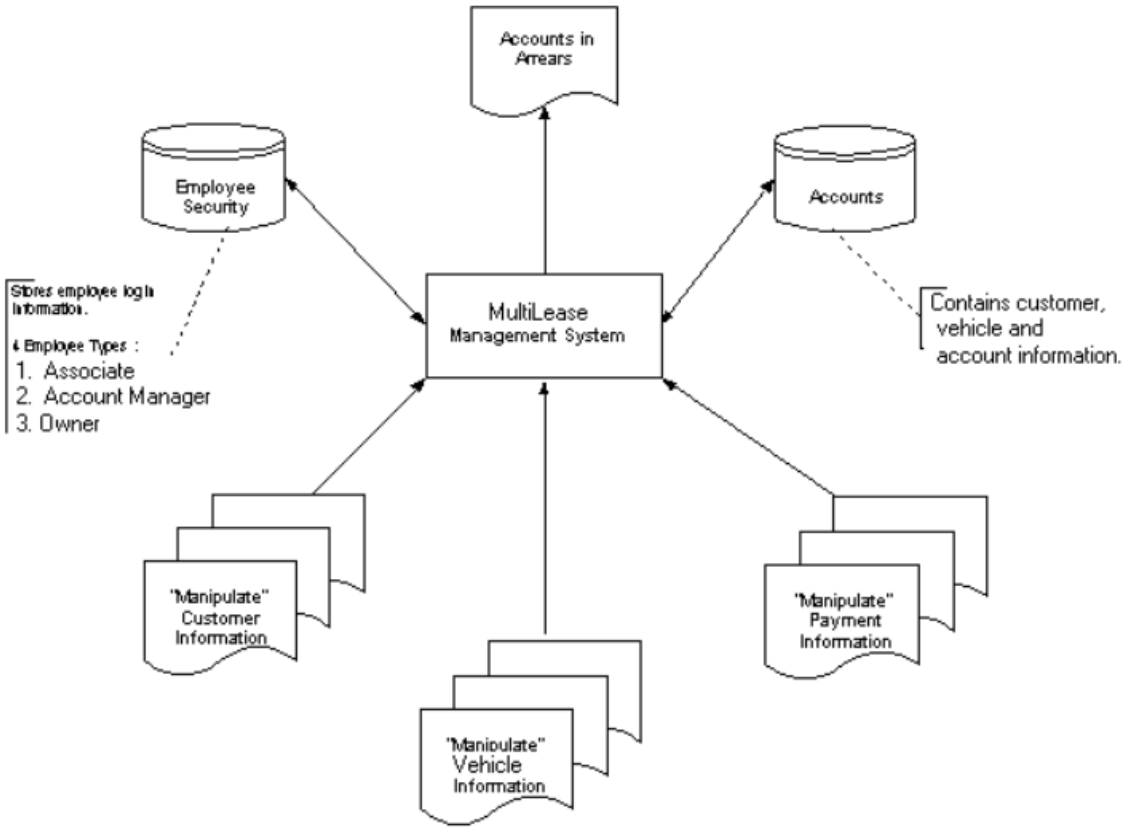

Figure 1: System flowchart for the MultiLease Management System

In this proposed system, Sales Associates will complete the initial paper-based contract, recording vehicle particulars between the customer and MultiLease. Following the contract acceptance, Sales Associates will be responsible for inputting the new customer information, vehicle inventory information and new lease payment information, contained in the original paper-based contracts and documents, into the system.

Account Managers will maintain customer relationships by collecting lease payments, collecting overdue accounts, dealing with customer concerns or complaints and terminating the lease contract when the car is returned at the end of the lease.

Mike Lee, the owner, will be able to perform all functions.

Data Requirements

This section describes the databases and what information must be stored in each.
The MultiLease Management System consists of the following two databases:

- ML_Employee database
- MultiLease database

Both databases exist on one server.

Data Rules

The following business rules are required in the MultiLease database:

- Typically leases are 1, 2, 3, or 4 years.
- Number of monthly payments on a lease range are 24, 36 and 48 months.
- Vehicles cannot have power locks if they do not have air conditioning.
- The first payment on a lease must be after the contract date.
- No lease should allow a customer to drive more than 150,000 kilometres without paying a premium. In other words, the maximum distance permitted for any lease is 150,000 km. After that, customers *must* pay the premium charge (per kilometre.) Leases where the maximum kilometres are less than 150,000 km are fine. Note: Customers pay premiums after they return their car at the end of the lease.
- The Vehicle VIN is unique.

The accounts database is described below.

The tables include:

- Customers
- Vehicles
- Leases
- Payments
- Audits

The database includes various lookup tables:

- Lease Terms
- Models
- Colours
- Types

The following information needs to be recorded about Customers:

- First and Last Name
- Address
- City
- Province
- Postal Code
- Phone Number

The following information needs to be recorded about Leases:

- Date the lease contract begins
- First payment date
- Amount of monthly payment
- Number of monthly payments
- Which vehicle the lease is for
- Which customer the lease is for
- The terms of the lease

The following information needs to be recorded about Vehicles:

- Vehicle VIN
- Model
- Type
- Colour
- Year
- Kilometres on odometer
- Whether or not the vehicle has been leased before
- Book value (current value of vehicle)
- Whether or not the vehicle has automatic transmission, air conditioning and power locks

The following information needs to be recorded about lease terms:

- Number of years
- Maximum kilometres
- Charge for extra mileage (cents/kilometre)

The Models, Colours and Types tables store available car models, colours and types respectively. Examples of car types include trucks, sedans (four door), and coupes (two door).

The following information needs to be recorded about payments:

- Date
- Amount
- Lease ID

The following information needs to be recorded about audits:

- Date
- Field information changed
- Change (old and new values)
- Lease ID

Database Usage

L'analyste de systèmes a découvert les faits suivants concernant l'usage des bases de données

- A search for an existing customer in the Customers table is by phone number. The user will want to know the customer's first and last name.
- Searches in Vehicles table typically involve a model and type.
- Searches in Leases table typically involve a customer and contract date.
- LeaseTerms, Models, Colours, and Types tables are rarely searched apart from a lease.
- Vehicle information does not change until a new model year. New customers and lease agreements are added continuously.

Payment Procedures

Payments entered incorrectly cannot be deleted. Instead, they are voided. A voided payment has a payment value of $0.00. A reason should always be given for a voided payment. By voiding a payment instead of deleting it, MultiLease ensures that it has an accurate record of all payments made, even if some of the entries were originally incorrect. Also, voiding payments prevents employee theft.

Auditing System

In order to keep track of amendments to lease agreements, Mr. Lee would like an auditing system implemented. The auditing system should record whenever changes have been made to the **Leases** table. Specifically, you should record when the changes were made, what fields were affected, and what their old and new values were.

Updates to Procedures

The systems analyst has identified procedural changes that require implementation:

Payment Procedures

Payment is due on the monthly anniversary of the contract date. Payments must be recorded in the database. Each payment requires GST and provincial taxes (use the tax rate in your province) be added. Payments more than 3 days late are considered past due. Payments that are past due incur a 2% interest charge. For example, if a payment of $350.00 is late, the customer now owes MultiLease $357.00.

End of Lease Procedure

At the end of a lease agreement, the customer is expected to return the car to MultiLease. The Account Manager who is responsible for the lease examines the car to ensure it is in good condition and checks the odometer. If the customer has exceeded their maximum kilometres (as specified on the lease), a *lease premium* payment must be made. The premium payment is equal to the number of kilometres over the maximum times the premium rate. For example, if a lease has a premium rate of $0.15 / km and a customer drove 5,000 km over the limit, they owe MultiLease $750.00, plus GST and PST.

Auditing System

At a later date, Mr. Lee would like to be able to run a report that lists changes made to an individual lease.
Note: You do not have to add this report for the purposes of this project. Implementing the audit system is sufficient.

Login and Security Issues

Two elements of security need to be addressed. Each user is assigned a loginID name, password, and group. Mike Lee wants the application's features and functions to be disabled depending on the user's security group. This type of security protects customer and merchant privacy. Table 1 identifies the program features that are assigned to each user group.

	Vehicles				Customers and Leases					Payments			
Group	**A**	**D**	**E**	**V**	**A**	**D**	**E**	**V**	**T**	**A**	**O**	**E**	**V**
Manager				***X***			***X***	***X***	***X***	***X***	***X***		***X***
Associate	***X***			***X***	***X***		***X***	***X***					***X***
Owner	***X***	***X***	***X***	***X***	***X***	***X***	***X***	***X***	***X***	***X***	***X***		***X***

Table 1: Program features and functions available to each user group

Key:

A Add	**V** View
E Edit	**O** Void
D Delete	**T** Terminate lease agreement (car returned)

Unauthorized users often attempt to access systems by guessing userIDs and passwords. The login system prevents hackers from breaking into the system this way by permitting only three simultaneous login attempts. After the third attempt the application closes and the user's loginID is recorded in a log file.

Program Design Requirements

You are responsible for modelling a major part of the system. You are also responsible for designing the necessary menus and forms for the MultiLease system. Users should be able to view, edit, add and delete information as described below. Users are not allowed to perform every task (see the security grid above.)

Account Managers should be able to:

- View and edit customers
- View and edit leases
- Add, view and *void* payments (**Note:** No one is permitted to edit a payment. If it is incorrect it must be voided and re-entered.)
- Terminate lease agreement (at the end of a lease when the car is returned)
-

Sales Associates should be able to:

- Add a new vehicle
- Add, edit, and view customers
- Add, edit and view leases
- View payments

Owners should be able to:

- Add, edit, view and delete vehicles
- Add, edit, view and delete customers
- Add, edit, view and delete leases
- Add, view and *void* payments (Note: No one is permitted to edit a payment. If it is incorrect it must be voided and re-entered.)

Systems Documentation

Since you will be working on only part of the system modelling, you must adequately document your work for other members of your team, or in the event you are unable to complete your portion.

Mr. Lee wants your assurance there will be no delays and additional expense. He has scheduled hardware installation, and training days for his employees. He understands there will be some disruption to his customers but insists it be very minimal.

Lease Company – Lease Agreement

Sold to:
Candie Wrapper
1000 Lollipop Lane
Halifax, NS
B1X 1X1
Phone: (902) 123-4567

Lease Vehicle:

2003 SC-430, VIN: 3W9T1-2Q10D-12D0P-2E1R2
km: 0 / book: $90,000

Transmission	Type	Options
☒ Auto	☒ 2dr Coupe	☒ Air
☐ Manual	☐ 4dr Sedan	☒ Power Locks
	☐ Truck	
	☐ SUV	
	☐ Van	

Exterior	Lease	New
☒ Deep Blue	☐ 1 year	☒ New Vehicle
☐ Racey Red	☐ 2 years	☐ Re-lease
☐ Lemon Yellow	☒ 3 years	
☐ Lime Green	☐ 4 years	
☐ Silver Grey		

Lease Terms

Contract Date: 2004/01/15
First Payment: 2004/02/15
Max. Mileage: 120,000 km
Premium: $0.25 / km
Monthly Payment: $650.00 / 36 payments

Your Task

You are responsible for only part of the solution's development. You will produce that part of the application for users to add a new lease, modify a lease, make and void a payment (this includes a down payment).

1. Planning

Application development is an iterative process. During an application's initial and subsequent development phases, it is usual to modify existing processes and discover new ones.

Review the designs you created in Systems Analysis and Design: Level 2 (SD2E) to add a lease and for the user interface. Consider the MultiLease database you created in Database Architecture (SQAE) and Database Design for SQL Server Server (SQDE).

Compare your initial design to the four requirements listed above under Your Task. Determine what additional procedures you need to complete the requirements. Update your LeaseVehicle design documents if required. You do not need to re-create the design documents. Use pencil or pen to note any changes, additions and rough 'quick sketch' design drawings. Determine additional classes based on the tables and their fields.

Review the storyboard, menu hierarchy and interface design. Decide what parts you need and if you need to make any changes to accommodate the four requirements (for example, data list controls), and any additional processes you identified. Use pencil or pen to update your interface design documents.

SYSTEM NOTE: For your part of the solution, you are responsible for leases, payments and audits only. You are not required to write code to add new customers or vehicles, or search for existing customers and vehicles. Instead, select from the information already existing in the database (for example from a data list), or add new customer and new vehicle records you need to add a new lease directly to the database through Visual Studio .NET's Server Explorer. For the payments report, you need to write a SQL statement.

INTERFACE NOTE - KEEP THE APPLICATION SIMPLE: Your part of the application requires a minimum of three forms. Be careful to keep the complexity low so that you can complete the interface in time.

It is now time for your instructor to check your progress. After you have finished updating your design documentation for the MultiLease Management System contact your instructor for a brief review.

 About 2 hours should be enough to complete this step.

2. Create the Windows User Interface

Based on your updated storyboard and interface design documents, produce Windows forms complete with any menus, toolbars, and controls. Add code to any buttons and menu choices responsible for navigating from one form to another.

Do not add any database controls or database code in this step however you can add any code not associated with the database.

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 (eg. Status bars).

Revisit your design documents and note any additional revisions you discovered you need to make. In the interface design sketches, note any noncritical feature not incorporated due to lack of time (if you document these changes properly, you will not lose any marks).

 About 3 hours should be enough to complete this step.

3. Process Database Information

Once you have built the application's 'shell' and are able to navigate through the forms, you can connect to the MultiLease database using appropriate data controls.

NOTE: Do not use database security.

Work with one form at a time.
Fill the data sets and bind the data sets to the data controls you are using, if any. Add code to manipulate the data sets. Use the stored procedures you created. You can create new ones using Visual Studio .NET if you wish. Add any additional code to make each form functional.

Make sure you test and add error-handling as you work. 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 complete but doesn't do anything.

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

 About 14 hours should be enough to complete this step.

4. Write a conclusion

In a short paragraph, describe how adjustments to your design to accommodate the application development help build a business solution to address Mr. Lee's needs.

 About ½ hour should be enough to complete this step.

Améliorations

If you have time, you may wish to add some of the following enhancements to your project. You will not be awarded any additional marks for adding these enhancements, but completing them increases your knowledge and expands your skill set by giving you additional practice.

- Add a form to manage **Log Into System** (do not use Database Security).
- Incorporate **Database Security** (this will take some research).
- Add a form to **Return Vehicle**.
- Add a form to **Purchase Vehicle**.
- Add a form to **Add New Customer**, **Find Customer** and **Delete Customer**.
- Add a form to **Update Employees**.

MARKING SCHEME

You are graded on the following components:

Project component	Points
Updated System Modelling	**4**
• Submission includes Lease Vehicle documentation present showing changes	1
Design notations, flowcharts and sketches for additional\modified requirements.	3
Updated User Interface Design	**4**
• Submission includes storyboard documentation showing updates based on dialog design.	1
• Updated Menu hierarchy and Sketches of applicable Windows forms designs. Any planned Form control and features omissions noted.	3
System and Interface Design	**24**
• Windows forms and functionality meet specifications noted in design requirements.	11
• System and functionality meet specifications noted in design requirements	13
Processing Database Information	**60**
• Correct and working coding/binding of database components	30
• Use of appropriate ADO.NET controls including adapters, datasets, etc.	10
• Use of appropriate Windows form' data controls	10
• Appropriate internal documentation (comments)	5
• Use of appropriate error handling	5
Conclusion	**4**
Presentation Quality (presence of all required elements including title page, neatness)	**4**
Total	**100**

WHAT TO SUBMIT

For full marks, you must submit the following items:

- ✓ A title page, including your name, student number, instructor's name, and course name in an appropriate cover.
- ✓ Updated technical requirements (sequence diagrams, hierarchy charts, etc).
- ✓ Updated sketches of all storyboards and forms.
- ✓ A diskette containing all source code, as well as all other files required for the proper functioning of your application.
- ✓ A list of information required by your instructor to facilitate the execution of the application.
- ✓ Conclusion - Include your explanation of how your design modifications address Mr. Lee's business problem.

PENALTIES

- Late submissions receive a penalty of 5% per day.
- Projects that are more than three days late can be submitted for a maximum grade of 60%.
- Projects contaminated with a virus must be resubmitted and will receive a maximum grade of 60%.