Course title	Data processing Project 1 : MultiLease management system
Project code	420-TDD-ID v1-0 Project 1
Project title	MultiLease management system
Pages	6 pages (plus cover page)
Released	20/05/2020
Revised	20/05/2020

Data processing

MultiLease management system

INTRODUCTION

You are an application developer 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, the owner of MultiLease 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 WPF Windows forms
- Write programming code based on structured and object-oriented concepts.
- Access and manipulate a SQL Server database through an application
- Debug and handle errors to produce an error-free application.

TIME REQUIRED

You are given 15 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 three days.

MATERIALS REQUIRED

The following software is required:

Software

- Microsoft Visual Studio
- Microsoft SQL Server

Business Requirements Section

General Requirements

You will use the MultiLease database that you created in the previous course.

Program Design Requirements

You are responsible for modelling a major part of the system. You are also responsible for designing the necessary forms for the MultiLease system. Users should be able to view, edit, add and delete information as described below.

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** and **modify an existing lease**. You will also add a **login form** for users to access the application.

Step 1: Planning

Examine the structure of the database you have created in the previous course. Compare what you have with the demands of the application. Determine if you need to add anything to your database to accommodate these demands. You will probably have to add a new table to manage the user logins (username/password).

Design the user interface. Decide which controls you are going to use. For instance, you might want to use lists or comboboxes to minimize typing from the user. You could also use datepickers for the dates. Try to make it as easy as possible to use the app. **Ask your instructor to approve your designs and any changes you make to the database!**

Step 2: Create the forms

Use your interface designs to create the forms. The application should start with a login form. Make sure the application can navigate between the different forms.

Do not put any code related to databases in this step. Only add code to display the forms and navigate between the forms.

Do not put too much time on this step. Keep the interface simple. You can always go back and make it nicer later if you still have time.

Step 3: Process database information

When your application is structured and you can navigate between the forms, you can add the database code. Choose the connection method you wish to use. We recommend Entity Framework.

Work with one form at a time. Bind the appropriate controls to the database. Add all the code the make the forms fully functional.

Test the application and add the necessary error handling code. Keep in mind that an ugly application that works is better than a nice looking app that does nothing.

Enhancements

If you have time, you can add some enhancements to your project. They will not give you any points, but they are a good learning opportunity.

- Add a form to return a vehicle
- Add a form to buy a vehicle
- Add a form to create, modify and delete a customer

Marking Scheme

You will be graded on the following components:

Project component	Points
System and interface design	**30**
• Windows forms and functionality meet specifications noted in design requirements	15
• System and functionality meet specifications noted in design requirements	15
Processing database information	**70**
• Correct and working coding/binding of database components	20
• Database gets updated	30
• Login form functions correctly	10
• Use of appropriate error handling	10
Total	**100**

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%.