UNIVERSITY OF NORTH TEXAS
COLLLEGE OF ENGINEERING
COMPUTER SCIENCE & ENGINEERING DEPARTMENT

CSCE 5430 – Software Engineering
Spring 2023

Project1-Phase3

Due Date for Project1 Phase3: Friday March 24, 2023, at 5:00 pm

1 Overview

The analysis model represents the system under development from the user's point of view. The analysis model is consists of the analysis object model and the dynamic model. The analysis object model focuses on the individual concepts that are manipulated by the system, their properties and their relationships. The analysis object model, for iFANANCE has been developed during the activities in Project1-Phase2, where UML class diagrams is used. The dynamic model focuses on the behavior of the system. The dynamic model is depicted with sequence diagrams and with state machines.

In this Project1-Phase3, you will develop the dynamic analysis for iFANANCE. This will involve composing UML sequence diagrams, listing and describing operations. (You need to be familiar with the requirements in Project1-Phase 1 and 2, and the Appendix as well).

As you go through this Project1-Phase3, remember to keep up to date the project glossary and your project workbook revision history.

2 Goals

At the end of this Project1-Phase3, you should be able to:

- Verify an analysis class model.
- Identify and describe operations from models and/or from codes.
- Draw sequence diagrams from the analysis model and from existing codes.
- Understand and run the given CRUD GUI iFINANCE application that uses .NET C# and Entity Framework (added to the last part of Project1-Phase2 solution).
- Update the project glossary.

3 Realizing use cases

3.1 Pick use cases

In order to simplify this Project1-Phase3 and make your life much easier, you will pick some of iFANANCE use cases to realize. It's a good idea that all students start with the same use cases from iFANANCE use case diagrams. In particular, in this Project1-Phase3 we will build two sequence diagrams as follows:

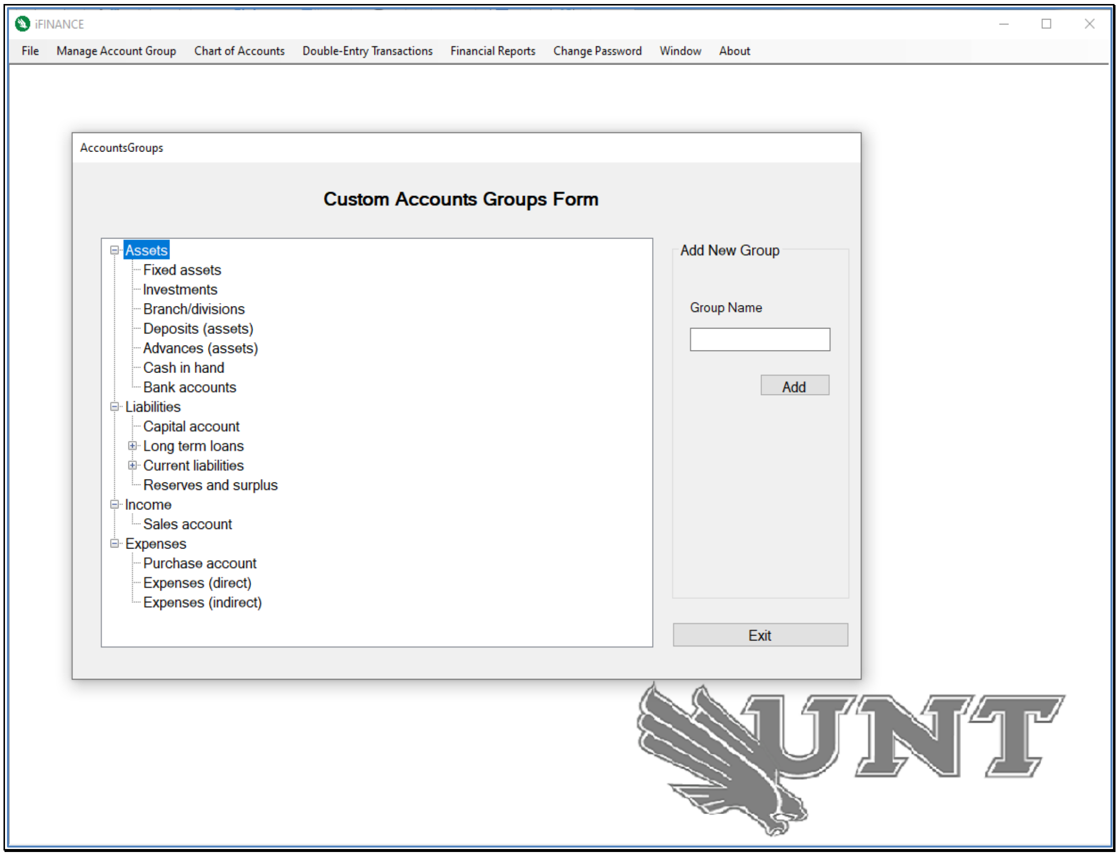
1. The first sequence diagram to realize the use case "Manage Account Groups"
2. The second sequence diagram to realize the use case "Manage Chart of Account"

Please develop your works based on the sample solutions given for Project1-Phase1 and Project1-Phase2.

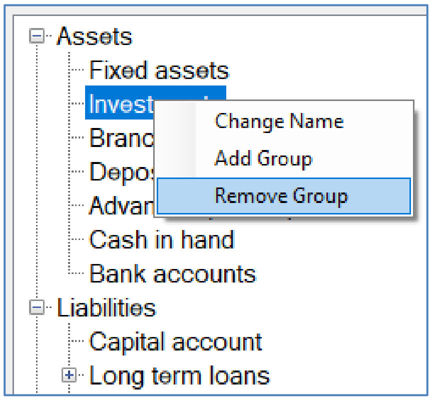
3.2 Manage Account Groups

Manage Account Groups use case has been fully implemented using C# and Entity Framework and it is given to you within the sample solution for assignment2.

1. This use case starts when the user chooses the "Manage Account Group" option from the main menu (shown below).
2. The Custom Accounts Group form is then displayed. The form shows a tree like view to display all existing custom groups and their sub groups, if any.
3. This form provides the user the ability to create, read, update and delete (CRUD) the system custom accounts groups.



4. All groups are divided into four categories, Assets, Liabilities, Income, and Expenses. Right click any of the existing group a context menu appears to show three options: change Name, Add Group, or Remove Group.
5. On add new group the cursor moves to the "Add New Group" text field in the right hand side of the form. Enter a new name and then click Add button.
6. This form has the full feature of displaying the groups and its sub groups as it saved in the database.
7. Note that, this use case has been implemented to fulfill the MVC architecture style.



Now, you need to read and get familiar with this given code. Examine the existing classes and see how they are communicating in order to satisfies the roles of the MVC architecture. This is an easy and important step that will help you answering the questions of this assignment.

3.3 Draw a sequence diagram

Use cases usually start with an actor telling a boundary object (a user interface) to do something or to retrieve some information. In such a case, you should draw the actor and the boundary with a line between them indicating the beginning of the interaction. This line should be decorated by a message being sent to the boundary by the actor. Proceed through the use case step by step, inventing message flows between objects to satisfy the use case steps.
The following tips may help you draw the diagram:

- 1st column should be the actor who initiated the use case
- 2nd column should be a boundary object (that the actor used to initiate the use case).
- 3rd column should be the control object that manages the rest of the use case
- Create a control object at beginning of event flow
- The control object can be created by boundary objects initiating use cases
- The control objects create the other boundary objects
- Ensure that the involved entity objects are in your class diagram.
- Realize all entity objects included in the given class diagram.
- Use abbreviations for parameters (if any) and then use notes to spell out the abbreviations.
- Show the Project1-Phase3 of return values only when they are not obvious (you will provide more detail in the operation descriptions).
- Name values only when they are used elsewhere in the diagram (for example, if a return value is subsequently used as a parameter).

1. Add a new section in your workbook titled ‘**Potential iFINANCE Sequence Diagrams’**.
2. Develop a UML sequence diagram for the given code of the “Manage Account Groups” use case and add it in this new section. All classes, the boundaries, the control, and the models (entities) along with the non-admin actor need to be included.
3. Develop another UML sequence diagram for the use case “Manage Chart of Accounts” and add it also to this new section.

Manage Chart of Accounts use case should work as follows:

1. The use case starts when the user chooses the “Chart of Account” option from the main menu.
2. The Chart of Accounts form is then displayed. The form shows a list like view to display all existing Master accounts information (one account per line) if such accounts exist in the MasterAccount table.
3. Each line of this list shows, the master account name, the opening amount, the closing amount and the custom account group this master account is belong to. Note that this custom account group should be chosen from the existing groups saved in the Groups table.

4. This for should provide the options of change the information of the existing master accounts, delete the existing master accounts, and/or add new master account into the list.
5. All data changes need to be reflected directly into the corresponding tables in the database.

Note that, the given code is very similar to the one you are required to develop or to present in a sequence diagram. The architecture is given, all operations to interact with the database for add, update or delete are given as well.

4 Detailing the operations

Use the following table to list each class you realized in the previous step (entities, boundaries and controllers), For each class, walk through the three sequence diagrams and record every message sent to an object of that class as an operation. For each operation, add a short sentence or two describing what the operation does, what its parameters are for and what it returns (if anything). As you do this, be sure to show the types of the parameters and return values (if any). Occasionally, you won't know the types yet or you won't want to commit yourself until design – in such cases, just ignore the operation type in your description. Bear in mind that parameter and return types should match the attribute types you chose in the object analysis model.

Class name	Receiving message (operation)	Brief Description

5 Dynamic Sequence Diagram Creation

Note that, the given code is very similar to the one you are required to create or to present in a sequence diagram. The architecture is given, all operations to interact with the database for add, update or delete are given as well.

Please carefully read **Part 8 the Grading** before and when you create your sequence diagram. You should include all the participants that have been mentioned in Part 8. You will use the given .NET C# project, which includes all entity classes (already created) and the corresponding database tables is also generated. **(If you have any question about the given code please feel free to ask me and/or any one of your TAs).**

6 Update the project glossary

As you go from one step to another in this work, new iFANANCE related terms (i.e., critical system operations, attributes, or states) may be discovered. Add these new terms accompanied with short descriptions to your glossary.

7 Hand in

By completing all the requirements in this Project1-Phase3, make sure you updated the table of content then submit your workbook through Canvas by the due date mentioned above.

8 Grading

CSCE 5430: Project1-Phase3 – marking scheme (100 marks)

Draw a sequence diagram that specify the following:

a) General for all sequence diagrams: (20 marks – 4 each)

1. 1st column should be the actor who initiated the use case
2. 2nd column should be a boundary object (that the actor used to initiate the use case).
3. 3rd column should be the control object that manages the rest of the use case
4. Create a control object at beginning of event flow
5. The control object cannot be created by entity objects

b) ManageAccountGroups SD: (40 marks)

1. The main boundary object is the ManageAccountGroupsForm (4 marks)
2. The control object is ManageAccountGroupsController (4 marks)
3. The following entity object(s) should be included:
   Category, (4 marks)
   Group. (4 marks)
4. The messages and events should be similar to the provided sample solution (20 marks).
5. The actor should be non-Admin user. (4 mark)

c) ChartOfAccounts SD: (40 marks)

1. The main boundary object is the ChartOfAccountsForm (4 marks)
2. The control object is ChartOfAccountsControl (4 marks)
3. The following entity object(s) should be included:
   MasterAccount, (6 marks)
   Group. (6 marks)
4. The messages and events should be similar to the provided sample solution (20 marks).