CIS 150, Summer 2008, Lab 11
Intro to Software Engineering, UML, and Use-Case Diagrams

Assigned: Mon, 8/11/08
Due: Mon, 8/18/08 (Note: there is no class or lab on 8/18/08, final exam is on 8/22/08)

1. Software Engineering is the development of large software systems and is usually divided into the following phases:

Requirement Analysis – understanding what is wanted and specifying what will be developed
Software Design – describing software solution and specifying how software will solve the task
Implementation – developing the program using a programming language or using tools
Maintenance – maintaining and upgrading the software

2. Unified Modeling Language (UML) is an object-oriented graphical language for visualizing, specifying, constructing, and documenting software systems. UML is commonly used for the development of large programs (software engineering).

3. The following nine types of diagrams are used in UML Version 2.0:

1) Use Case Diagrams
2) Class Diagrams
3) Object Diagrams
4) Sequence Diagrams
5) Collaboration Diagrams
6) Statechart Diagrams
7) Activity Diagrams
8) Component Diagrams
9) Deployment Diagrams

4. In all nine diagrams, concepts are depicted as symbols and relationships are depicted as lines connecting the symbols. A brief description of all nine diagrams is given below. The focus of this lab will be on *use-case* diagrams.

a) Use Case Diagrams describe the functionality of a system and users of the system.
b) Class Diagrams describe the static structure of a system, or how it is structured rather than how it behaves
c) Object Diagrams are similar to class diagrams, but describe the static structure of a system at a particular time.
d) Sequence Diagrams show the flow of a program in terms of selection, looping, and sequence structures; in other words, how the program executes.
e) Collaboration Diagrams show the flow of a program in a manner that facilitates an assessment of the distribution of processing between the different objects that make up the system.
f) Statechart Diagrams provide a way to model the various states in which an object can exist.
g) Activity Diagrams describe the activities of a class in response to internal processing.

h) Component Diagrams describe the organization of and dependencies among software implementation components such as source code, object code, and executable code.
i) Deployment Diagrams show the physical layout of a software system and where the various components of the system will reside.

5. More about Use Case Diagrams: Use-case diagrams describe the functionality of a system and users of the system. They contain the following elements:

1) *Actors*, which represent users of a system, including human users and other systems
2) *Use cases*, which represent functionality or services provided by a system to users

As an example, the use-case diagram for a miniATM (Automatic Teller Machine) is shown in figure 1 below. The miniATM provides three functions:

- Checking Deposit
- Checking Withdrawal
- Balance Enquiry

The *customer* actor is a user of the miniATM machine. The large rectangle represents the boundary of the system. The name of the system appears above the rectangle. The stick person icon represents an actor who is a human user. The *customer records* rectangle icon represents an actor that is not a human user.

The name of the *customer records* actor is preceded by an "<<actor>>" string. This string is called a *stereotype* and indicates that *customer records* is an actor. This stereotype and the stick person icon are equivalent. The stick person icon is usually used to represent humans, whereas the "<<actor>>" string is usually used to represent systems. The ellipsis represent use cases. The lines connecting actors and use cases indicate that the actors use or participate in the functionality provided by the use cases. Each use case can be further described by one or more additional use case diagrams.

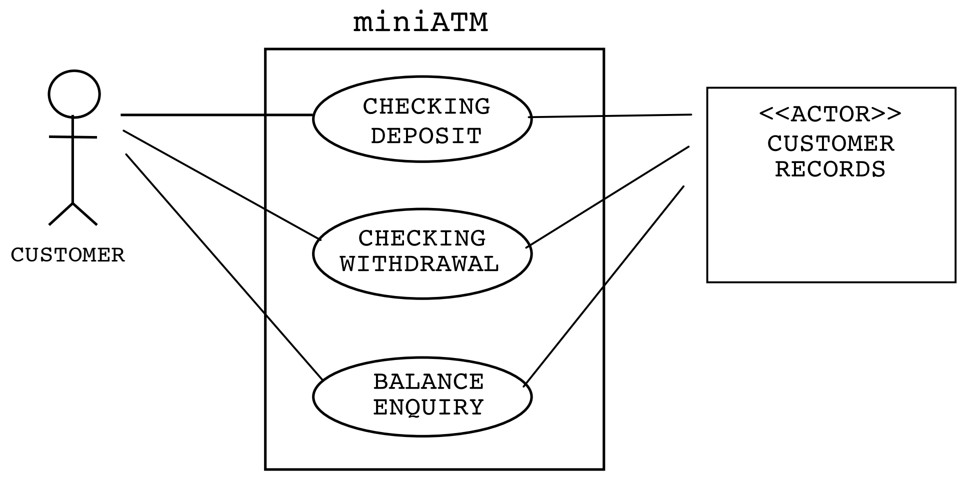

Figure 1. Use-case diagram for miniATM.

Lab 11 Exercises

Assigned: Mon 8/11/08
Due: Mon 8/18/08

Name:____

Answer the questions on the next two pages and turn in to your lab instructor or upload to assignment Lab11 at the CIS150SU08 VLT site.

1. Which UML diagram is most suitable for describing the functions to be provided by a program?

2. Which UML diagram is most suitable for describing the logic of a program?

3. Give the names of two UML diagrams that should be drawn for a program before the program is written in C++:

4. Give the names of two UML diagrams that could be drawn for a C++ program after the program is written:

5. Indicate whether each of the following is TRUE or FALSE:

_________ UML is an object-oriented language

_________ UML provides a use-case diagram and nine other diagrams

_________ *use cases* represent the functions provided by a software system

_________ *sequence diagrams* and *collaboration diagrams* provide the same information in two different formats

_________ in the use-case diagram of figure 1, the *customer* actor uses the miniATM and the miniATM uses the *customer records* actor

6. Add the following functions to the miniATM use-case diagram in figure 1:

   - Open New Account
   - Savings Deposit
   - Savings Withdrawal
   - Close Existing Account

   Draw the modified use-case diagram below: