MODELING SPECIFICATIONS

Presented by: Team Zodiac

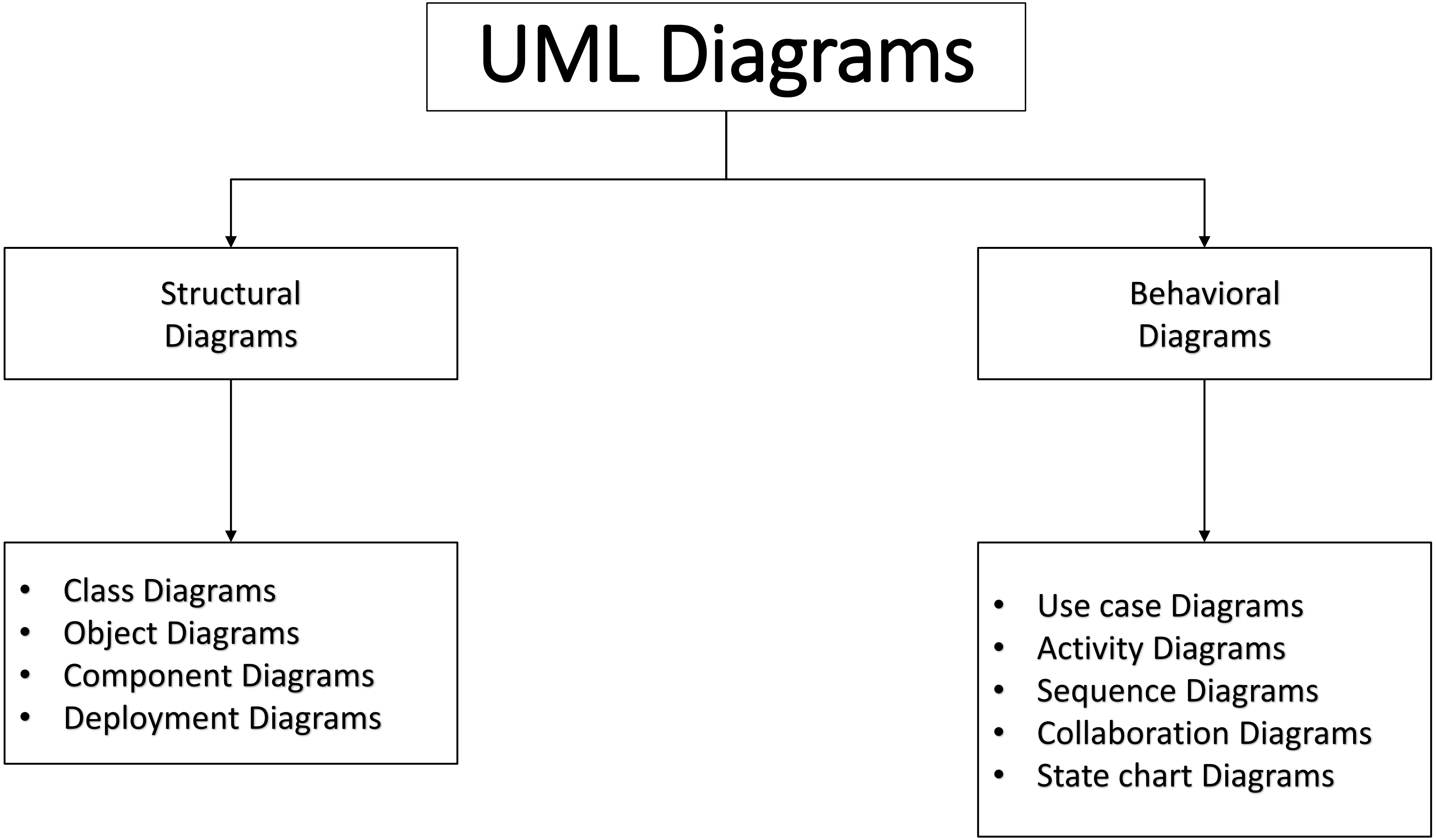
What are Modeling Specifications?

It is an approach to describing a software where various parts that make up the software such as its functions, its developmental phase and its description are represented via diagrams

The best and the internationally recognized standard for modeling softwares is known as the **Unified Modeling Language (UML)**

UNIFIED MODELING LANGUAGE (UML)

- Described as a pictorial language used to make blueprints which is geared towards providing a simple modeling mechanism for all practical systems
- Note that though it is used in software designing and planning, it is not a programming language.
- UML is merely a collection of diagrams that represent various instances of the static and dynamic aspects of a software
- Was adopted by the Object Management Group (OMG) as a standard for modeling software systems in 1997 and the current version is UML 2.5 (released in June 2015).



UML Diagrams

Structural Diagrams

- Class Diagrams
- Object Diagrams
- Component Diagrams
- Deployment Diagrams

Behavioral Diagrams

- Use case Diagrams
- Activity Diagrams
- Sequence Diagrams
- Collaboration Diagrams
- State chart Diagrams

STATIC OR STRUCTURAL DIAGRAMS

- Used to represent the static aspects of the software system; the parts which are set-up as the backbone of the software system.
- These static parts do not usually change with interactions and are often referred to as the coded aspect of the software system.
- Commonly used structural Diagrams are the class diagram, the object diagram and the component diagram

BEHAVIORAL OR DYNAMIC DIAGRAMS

- Simply illustrates the behavior of the software system or used to describe the functionality of the system
- The dynamic aspect of the software system is described as the ‘moving part of the system’, which parts vary as the system is being interacted with
- Commonly used behavioral diagrams include Use-Case Diagrams and Activity Diagrams

CASE STUDY: A banking software developed by Team Zodiac

FEATURES OF THE SOFTWARE

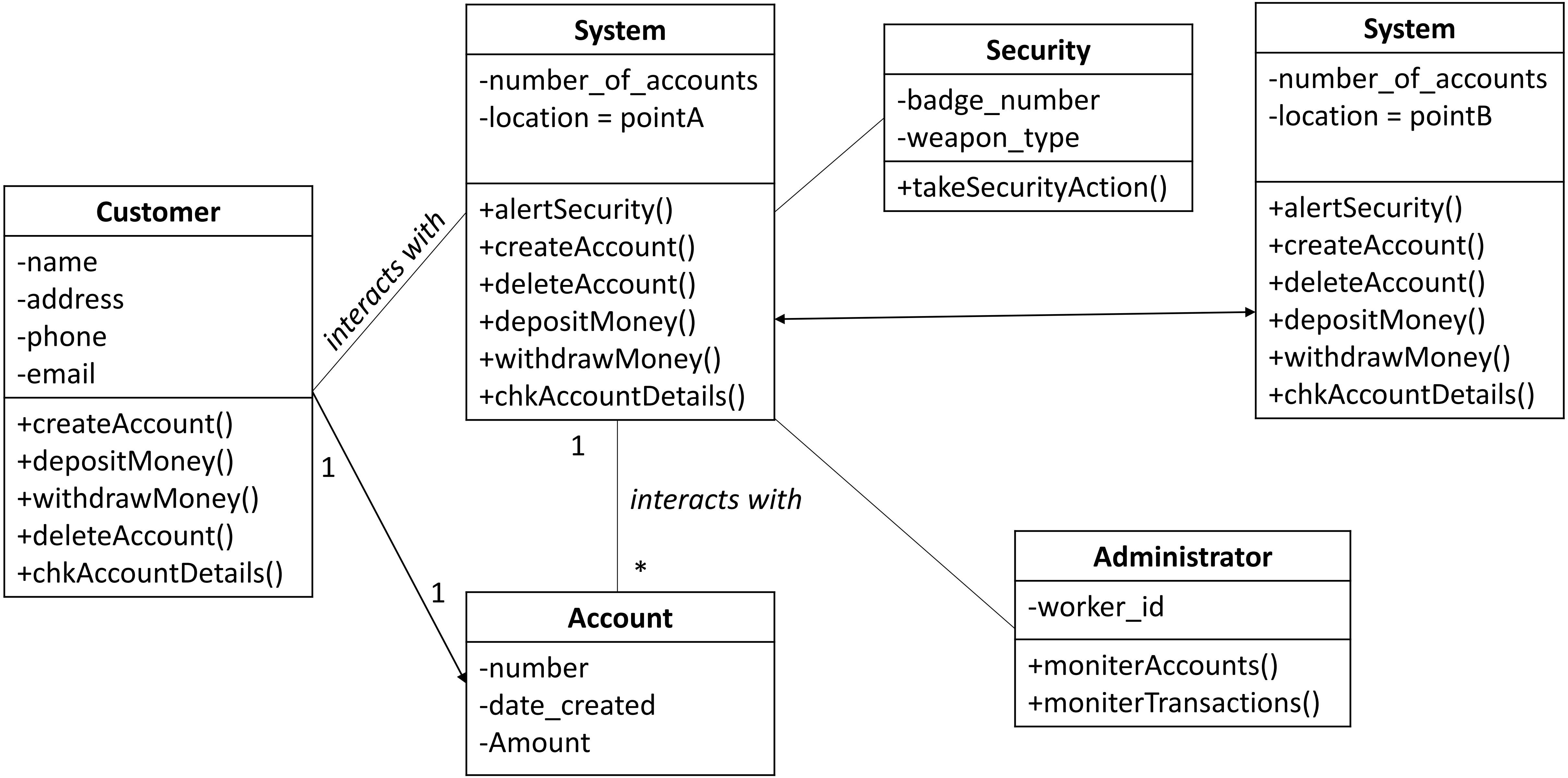
- Allows a customer to create an account.
- Customer can have only one account
- Allows a customer to access his/her account details via a properly encrypted 4-PIN number.
- Allows a customer to withdraw money and add money to his/her account.
- Allows the administrator to monitor all accounts created by all customers
- Customer can check his/her account details
- Allows other branches of the bank to have access to the customer's account details
- Dispatches security when a customer's PIN number is stolen or the account authentication fails 4 times
- Allows customer to delete his/her account

CLASS DIAGRAMS

- A type of static diagram
- Describes the structure of the software system using its classes, attributes and methods of a particular class and its interaction with other classes and even objects.
- Termed as the main-building block of object-oriented modeling, meaning it is prevalent in object-oriented designing.
- A class in a class diagram is represented as a 3-columned block:

Class Name (Center Aligned and Pascal-cased)
Class Attributes (left aligned and lowercased)
Class Methods (left aligned and camel cased)

CLASS DIAGRAMS USING OUR CASE STUDY



System

-number_of_accounts
-location = pointA

+alertSecurity()
+createAccount()
+deleteAccount()
+depositMoney()
+withdrawMoney()
+chkAccountDetails()

Security

-badge_number
-weapon_type

+takeSecurityAction()

System

-number_of_accounts
-location = pointB

+alertSecurity()
+createAccount()
+deleteAccount()
+depositMoney()
+withdrawMoney()
+chkAccountDetails()

Customer

-name
-address
-phone
-email

+createAccount()
+depositMoney()
+withdrawMoney()
+deleteAccount()
+chkAccountDetails()

Account

-number
-date_created
-Amount

Administrator

-worker_id

+moniterAccounts()
+moniterTransactions()

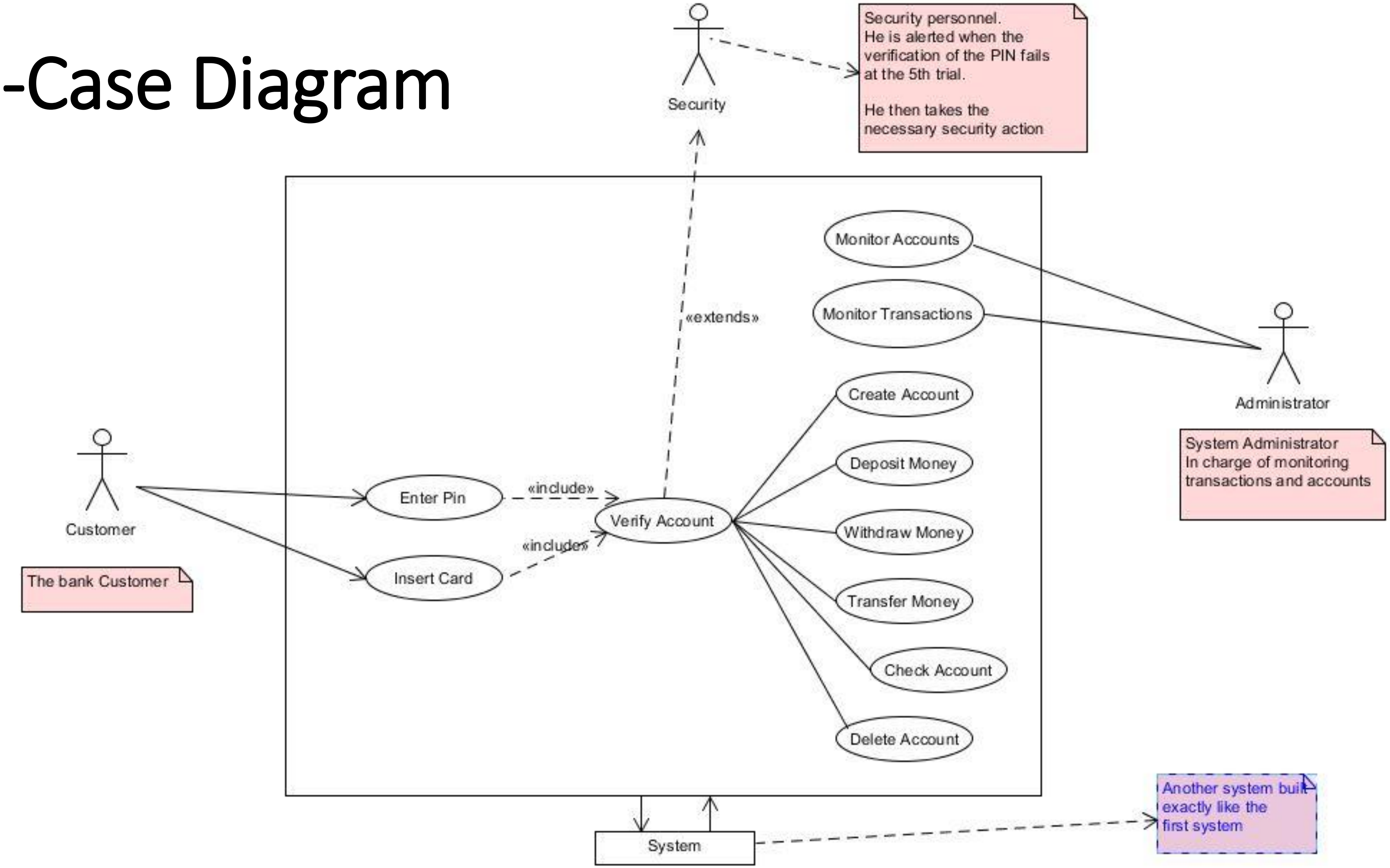
USE CASE DIAGRAMS

- An aspect of UML diagrams which captures the dynamic behavior of a software system.
- It shows **actors**, **use cases**, and the relationships existing between them and the software system as a whole.
- It seeks to identify internal and external factors influencing the system.

COMPONENTS OF A USE CASE DIAGRAM

- Actors
- Use Case
- Notes
- <<include>> and <<extends>>
- Communication lines
- System Boundaries

Use-Case Diagram



Security personnel.
He is alerted when the verification of the PIN fails at the 5th trial.

He then takes the necessary security action

System Administrator
In charge of monitoring transactions and accounts

The bank Customer

Another system built exactly like the first system

ACTIVITY DIAGRAMS

- It shows a sequential flow of control in the software system.
- It spells out how the system will flow with every user interaction, and other ‘else’ conditions that may occur
- It generally describes the system in very great detail, using some of the symbols from Use-Case Diagrams as well as a few others.

ACTIVITY DIAGRAM

THANK YOU