

Object Oriented Design

There's more...?



Outline

- **Object Oriented Design**
 - Identify the Classes
 - Identify what Information each Class Needs
 - Identify what each Class Needs to Do



Software Development Life Cycle

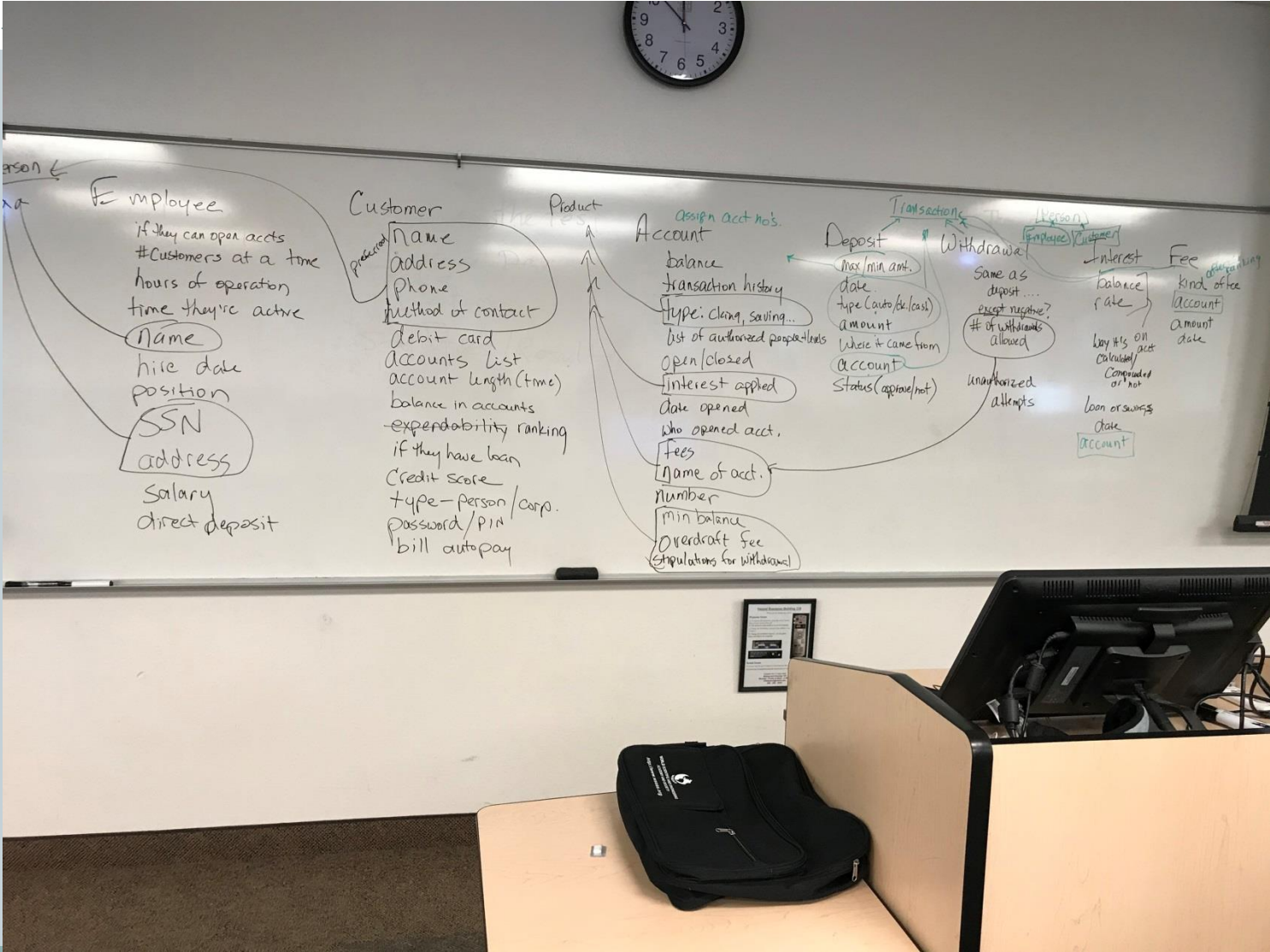
1. Understand the Problem = Requirements Analysis
2. Work out the Logic = Design
3. Convert it to Code = Implementation
4. Test/Debug
5. Maintenance

Today we will talk about requirements analysis and object oriented design.

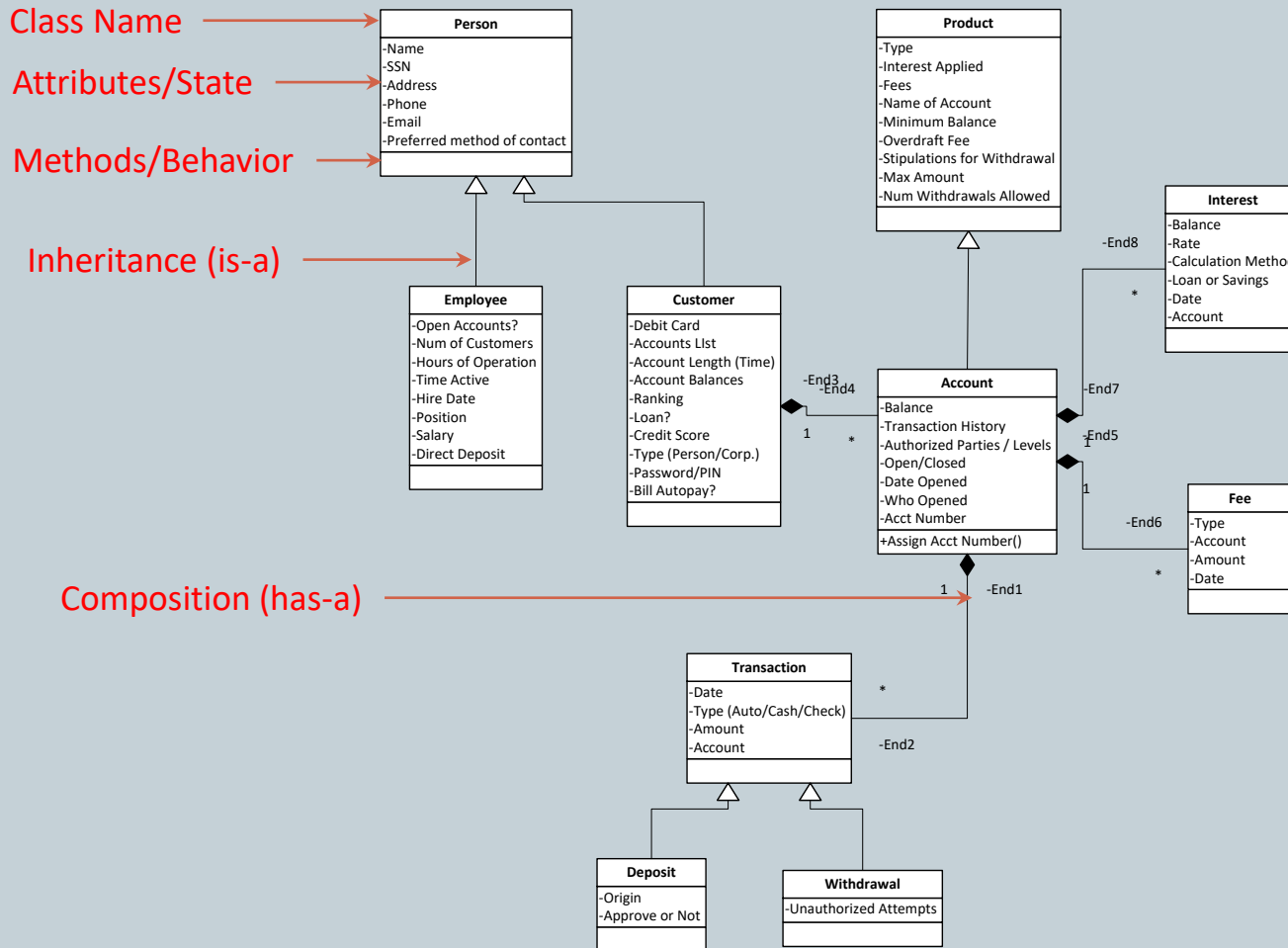
What are the Nouns?

- You have been hired to automate bank operations for a local credit union. They have told you that their business operates as follows:
 - Customers can open accounts. They can make deposits and withdrawals and can close accounts also. On some accounts interest needs to be added, and sometimes fees are deducted.
 - All employees can help customers with deposits and withdrawals. Only some employees are authorized to open and close accounts.

From Last Year...



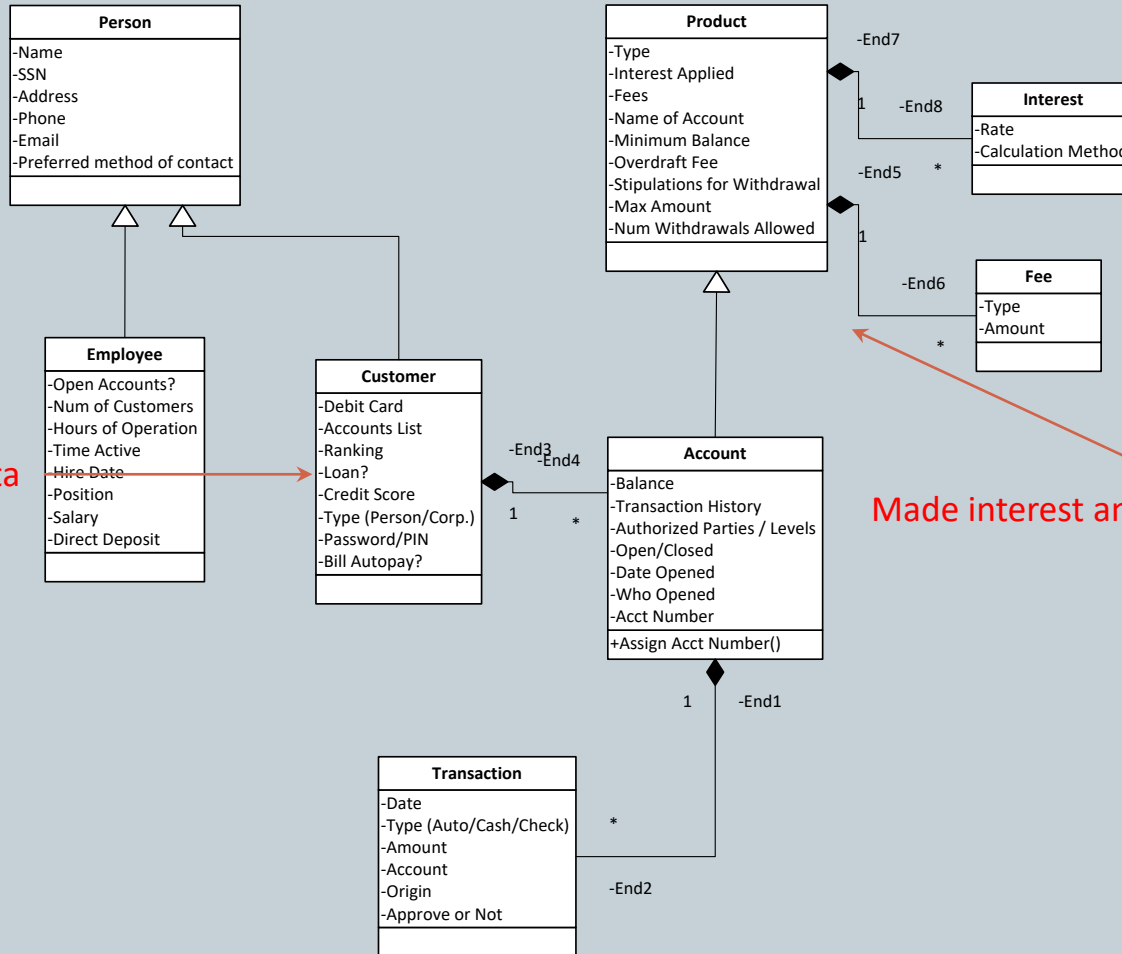
UML Diagram



Simplifying the Design – Classes and Attributes

- Look for repetition of data
 - Try to have each piece of data in only one place
- Look for “modifiers”
 - These might indicate the attribute should be in a different class
 - ✦ e.g. Under Customer, we have “Account Length (Time)” and “Account Balance”
 - Since they both reference account, they should probably be in the Account class
- Walk through each attribute and see if it makes sense
 - Does it really applies to that class

Modified UML



Removed account data

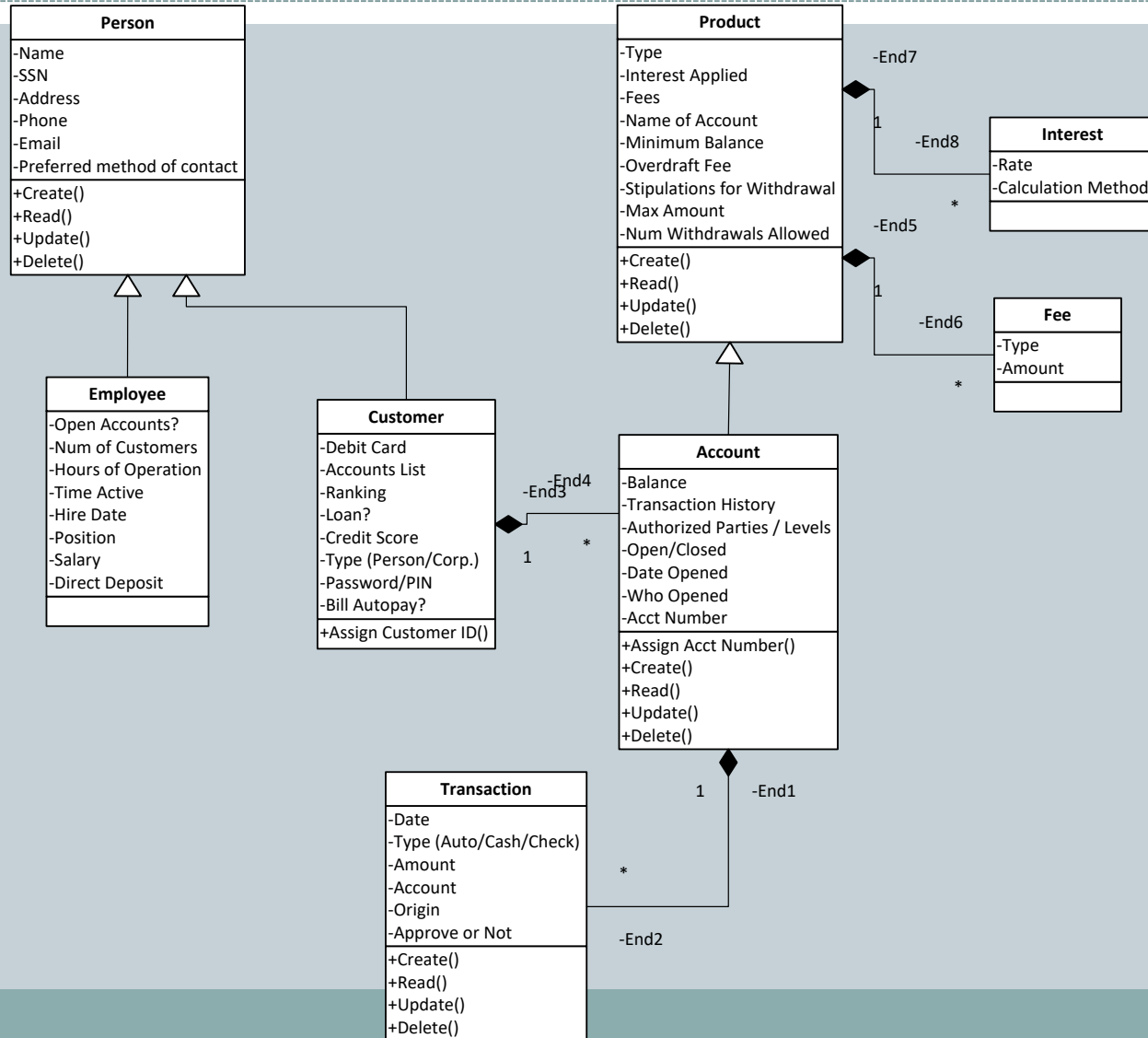
Made interest and fees part of product

Removed deposit and withdrawal

Add Behaviors - What are the Verbs?

- You have been hired to automate bank operations for a local credit union. They have told you that their business operates as follows:
 - Customers can open accounts. They can make deposits and withdrawals and can close accounts also. On some accounts interest needs to be added, and sometimes fees are deducted.
 - All employees can help customers with deposits and withdrawals. Only some employees are authorized to open and close accounts.

UML with Behaviors



More Design

- Use Cases
 - Walk through typical uses of your software and make sure the state and behavior support those cases
- Application Program Interface – API
 - Write an API for the interface to each of your classes
 - ✦ For each method, define:
 - Name
 - Input Parameters
 - Return Values
- Define data types for each attribute
 - Might mean splitting a single attribute into several

Implementation

- **Once we are happy with our class definitions, then we get to write some code!!**

Summary

- **Object Oriented Design**

- Identify the classes
- Identify what information each class needs
- Identify what each class needs to do
- Identify use cases
- Define the API
- Define the instance variables
- Finally – write some code!

