

CRC - Class Responsibility Collaboration

List of classes - each class written on a card

For each class -

You list down responsibility of that class

Which classes it is going to use/interact (collaborators)

=====

1. Identify classes (nouns)

2. Identify attributes (properties, descriptive phrases)

3. Identify operations (verbs, verb phrases)

4. Identify relationships (Dependency

— — — —>Inheritance. Weakest to strongest)

Express the design using UML (class diagram, sequence diagram, state diagram etc.)

=====

List of NOUNS -in the requirements will be

a good candidate for Class names
Which ones have significance?

Your responses for #1:

Bank containing many accounts 1 to many relationship

BankDatabase,
User/Customer,
Account

KeypadInputManager

CashDepositManager

CashDispenser

GUIManager/Display System

ATM - Class to maintain the overall balance(using its CashDispenser) in the atm machine & other details of the machine

- ***address/location/uniqueID***

UserTransactions -

View Balance, Cash withdrawal,

DepositFunds

Image processor (validating check deposits)

currentBalance (probably attribute of Account)

Account Credential Validator (probably an operation on Account)

Suggested features:

ChangePIN

Transfer

=====

Composite->contains->components

Composite -has-a- Component

Composition - Exclusive ownership of components(Not shared), each component is ***part-of*** one container

Lifecycle of component is within the lifecycle of the container - component objects do not exist without the container

object

Composite is going to have at least one instance of the component - zero instances are not allowed

One ATM object represents one ATM machine

=====

==