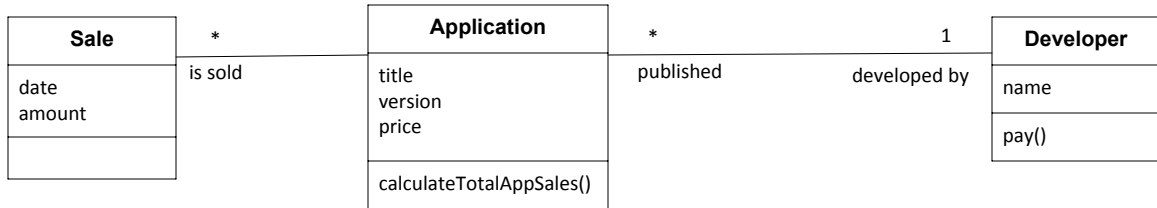


## Mobile App Store Design Example

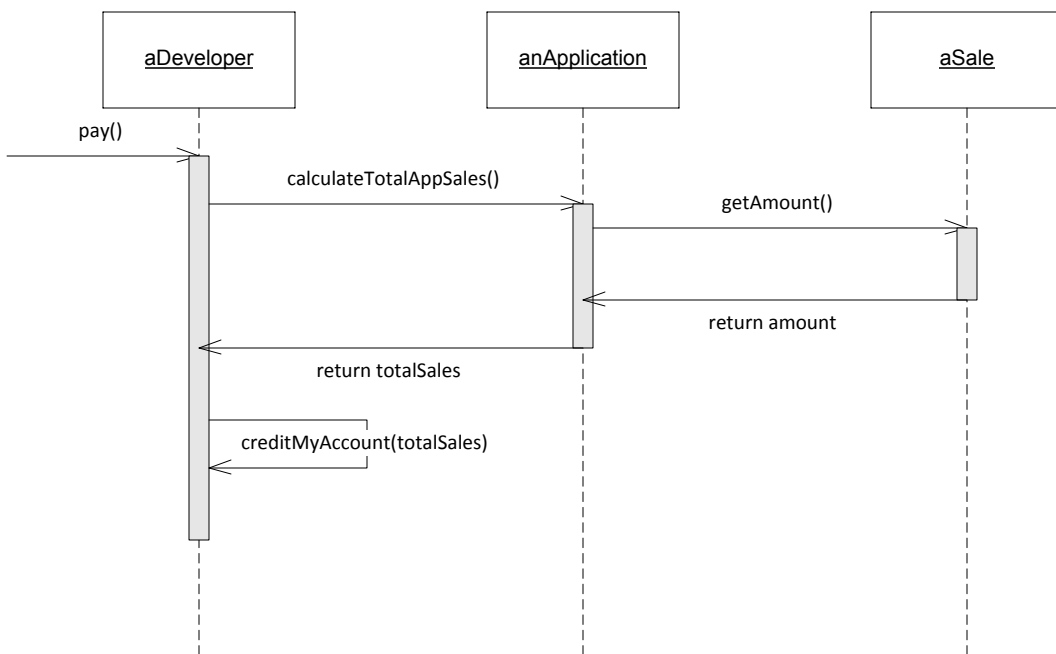
Designing a mobile app store – the developer interface.

- 1) Identifying key concepts: Developer -> Application(s) -> Sales (hopefully! ;)
- 2) Considering a scenario: Developer gets paid for the sold Apps

All diagrams are not strict UML and only show what we are focusing on, e.g. assume the presence of accessor methods such as `getAmount()` for present attributes

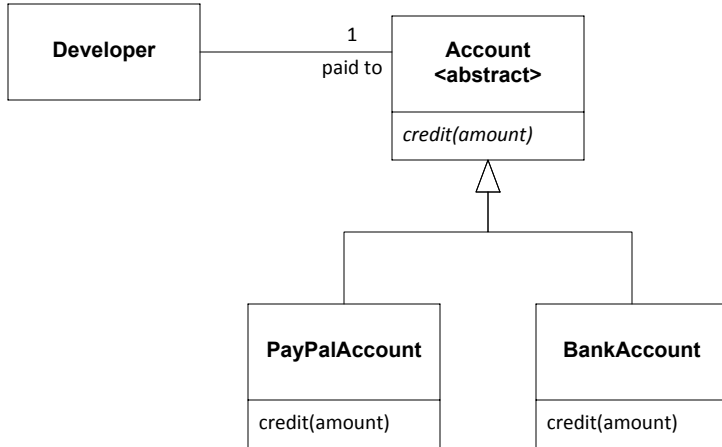


Scenario: aDeveloper published 1 application that was sold once

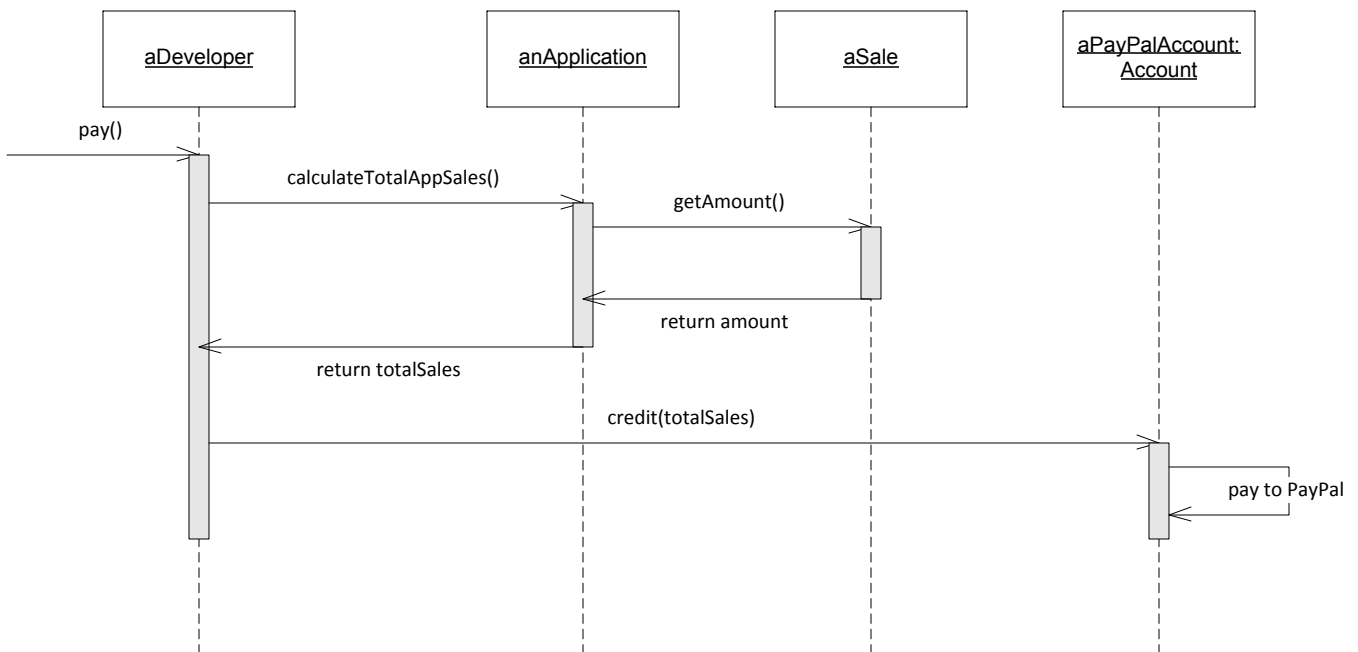


Observations: Developer class needs to know how to e.g. credit a bank account. What if we also wanted to support payments to PayPal accounts?

Let's decouple payment functionality from Developer -> introducing the Bank Account concept. To support multiple types of Accounts, it's useful to generalise the Account leaving the subclasses to implement the specific credit(amount) methods.



Scenario: aDeveloper with a PayPal account, published 1 application that was sold once

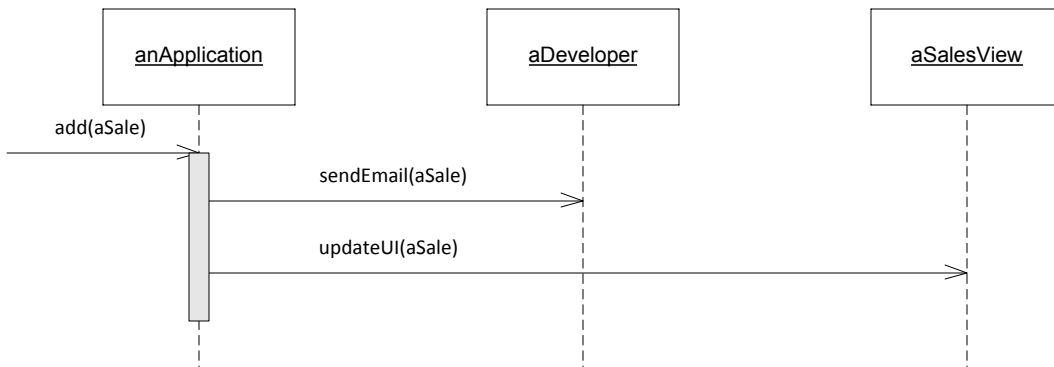


Introducing new requirements, when a sale is made:

- 1) the developer should be notified by email
- 2) an analytics view showing all app sales should be updated

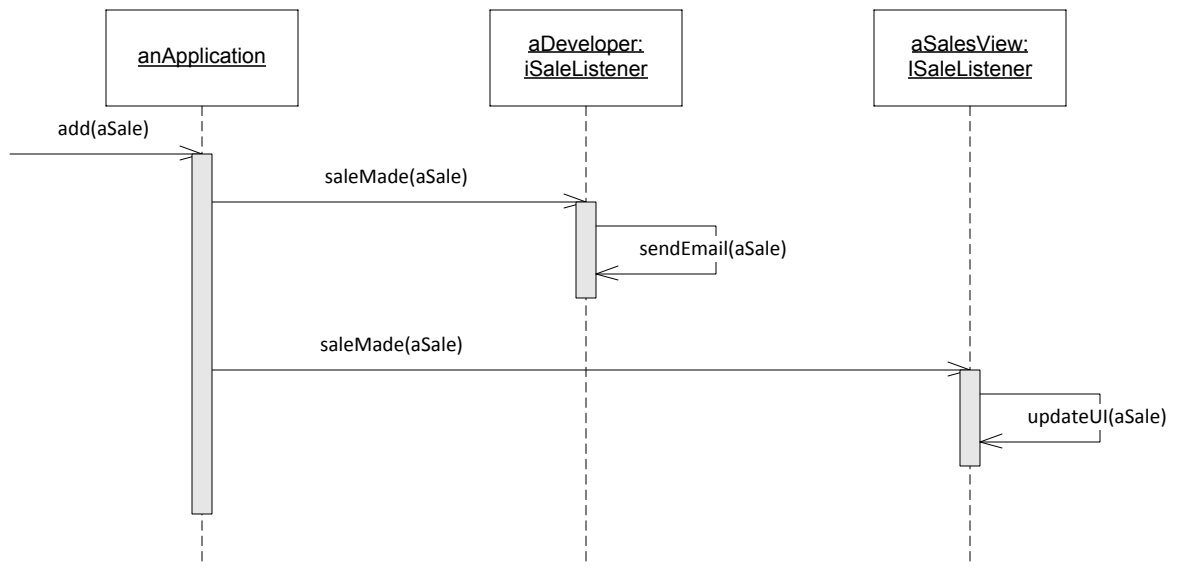
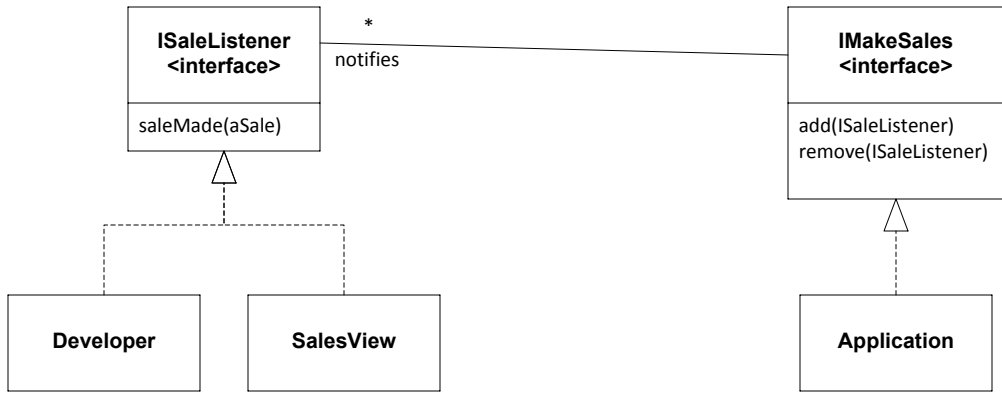
In the current system, anApp maintains a list of its own sales and has add(aSale) method.

A simple approach – perform required actions when a sale is added to the list.

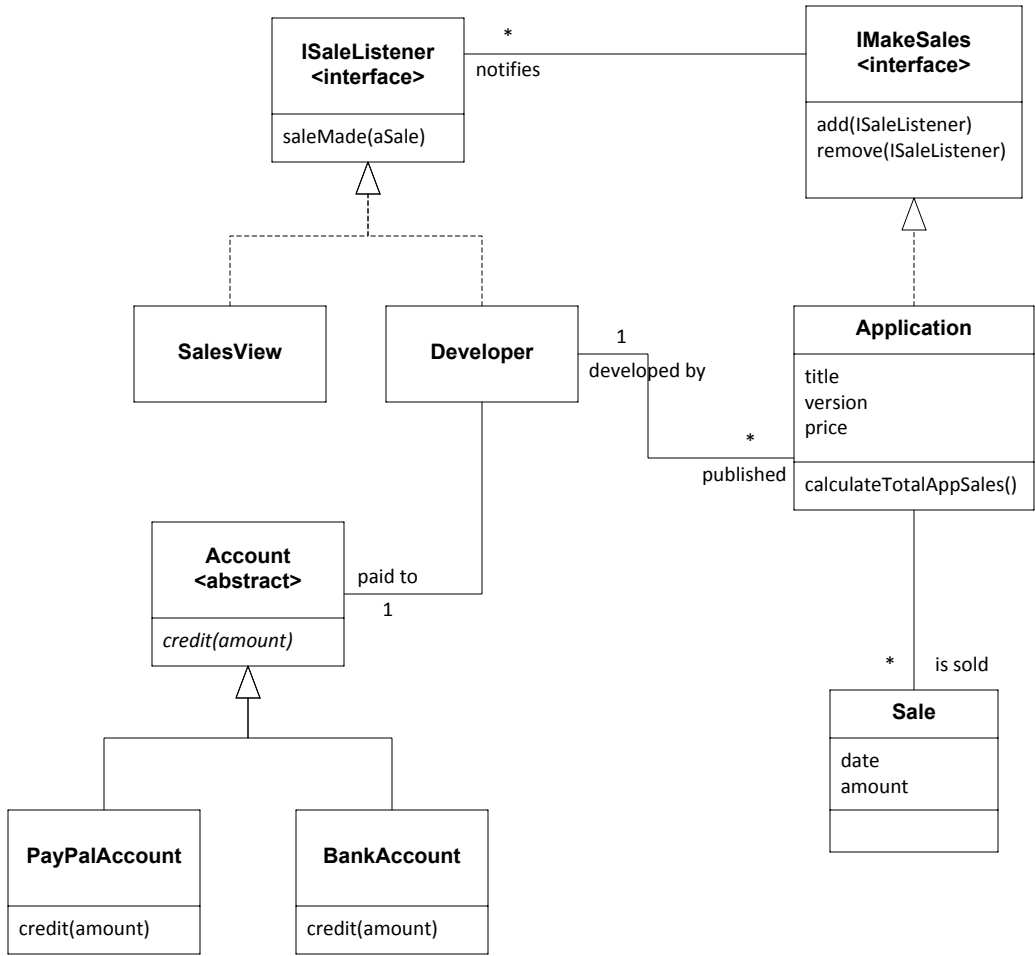


Observation: the Application needs to know all the actions that has to be done when a sale is made. Consider if more actions need to be added, or changed, e.g. send an SMS instead or in addition to the email.

Abstract the sales made notification functionality in to a set of interfaces – introducing the Observer design pattern.

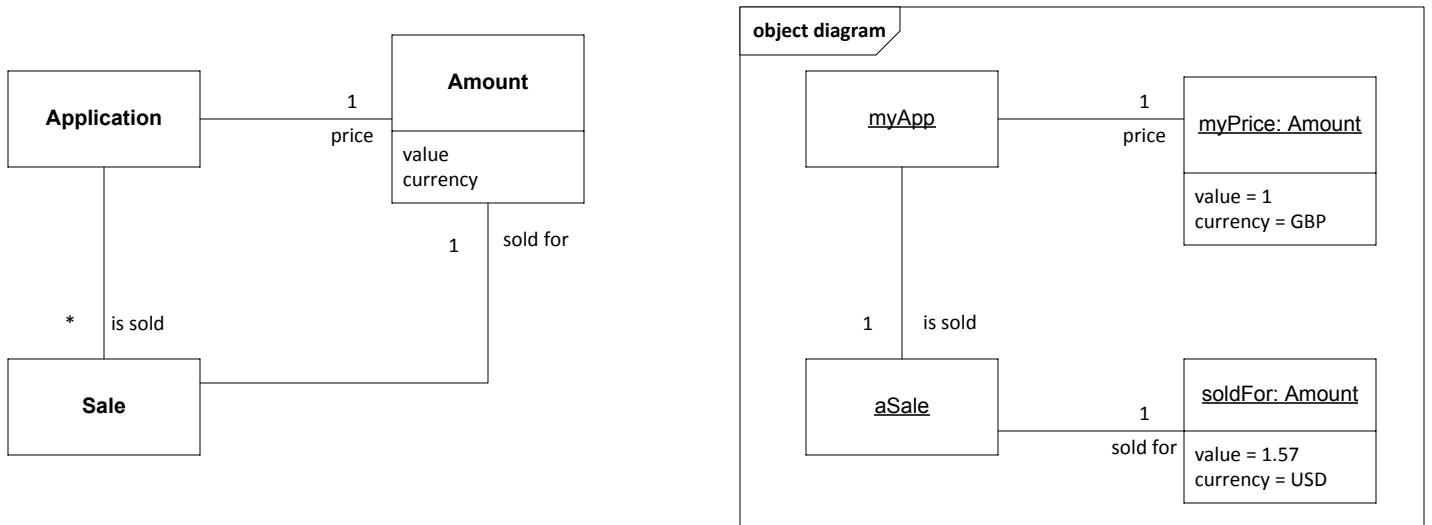


# Mobile App Store Design Example Overview



Another requirement – support sales in different countries/currencies.  
 Introducing the concept of Amount to encapsulate both the value and currency.  
 An Application has a Price (developer preferred currency), whereas a Sale has an Amount the App was sold in a particular country (and would be in the currency of that country).

Using Object Diagram to show two different instances of the Amount class.



Scenario: aDeveloper with a Bank Account, published 1 application (priced in GBP) that was sold once (in USD). The currencyConverter has a helper method to convert a given amount in a given currency.

