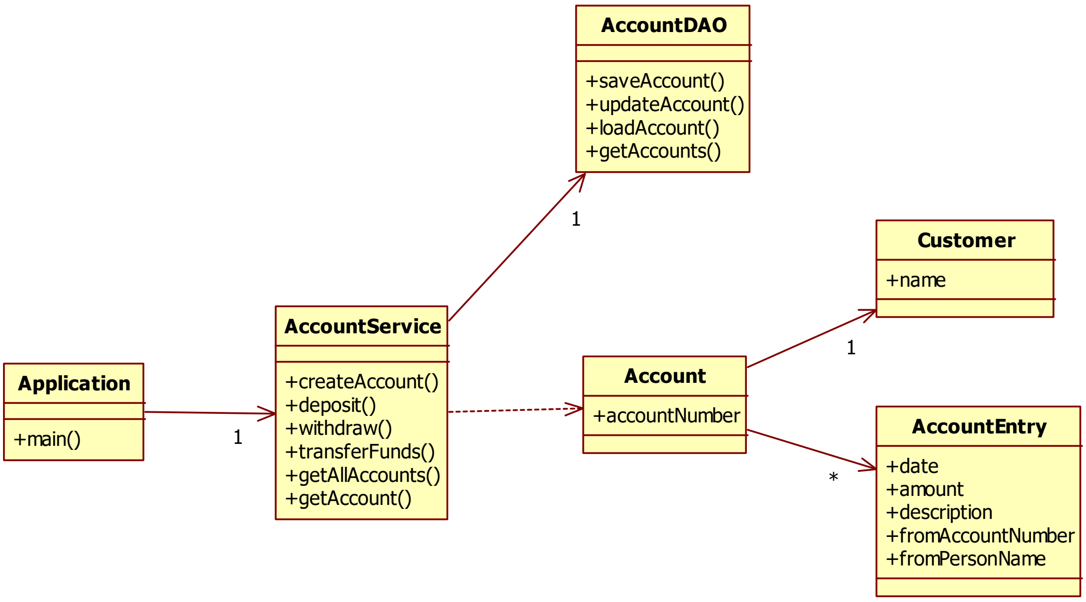
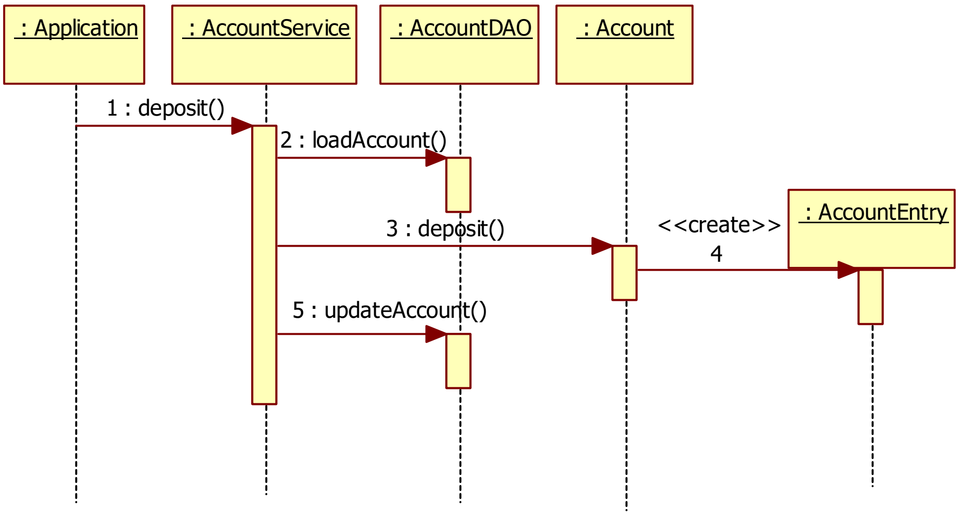
Lab 9

a. Factory pattern

Given is the following bank application:





Write an abstract factory class with the following functionality:

1. Whenever the amount of an account changes, we send an email to the customer (simulate with System.out.println()).
2. For testing, we want to use a MockAccountDAO and a MockEmailSender
3. For production we use the AccountDAO and en EmailSender class
4. We specify the environment (production or test) in config.properties file.

b. Builder pattern
   Modify the code so that we use a Builder to create the AccountEntry objects

c. Singleton pattern
   Modify the code so that the EmailSender is a singleton that is thread safe

d. Decorator pattern
   Decorate the AccountDAO with an AccountDAOLogger that logs all actions we perform on the AccountDAO.