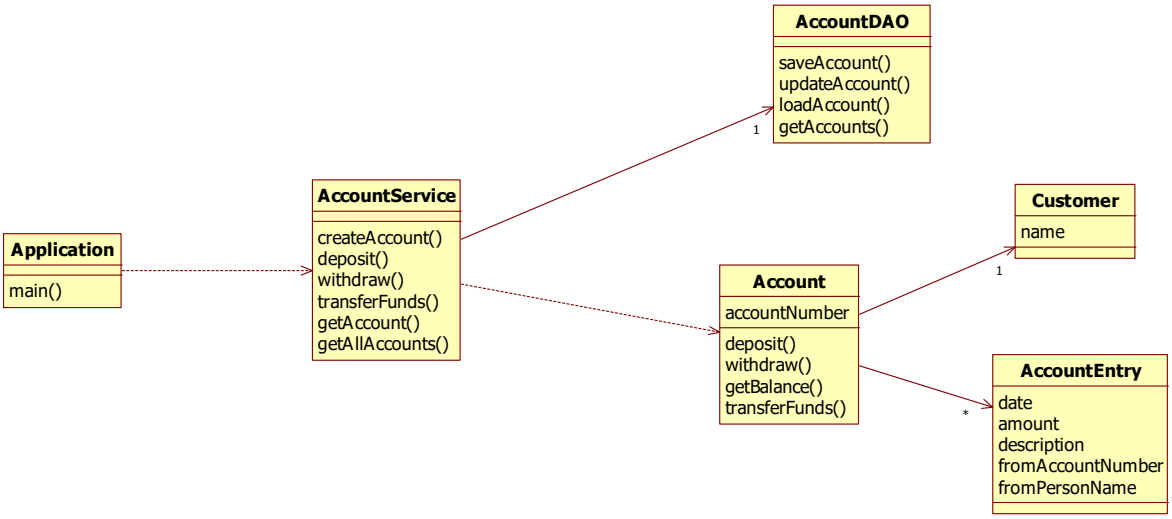
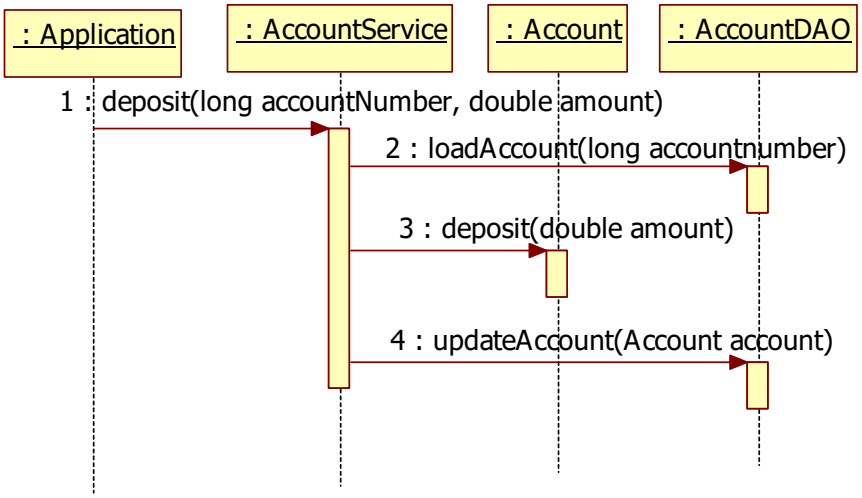
Lab 8

a. Given is the following bank application:

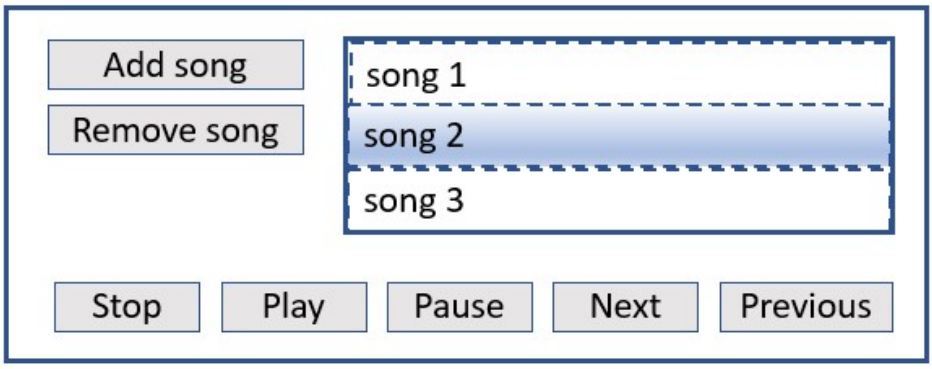




a. Add the following dynamic proxies
   1. Write a logging proxy that logs all method calls on the AccountDAO class.
   2. Write a timing proxy that computes the time for every service level method we invoke
   3. Apply the timing proxy also for DAO method calls.

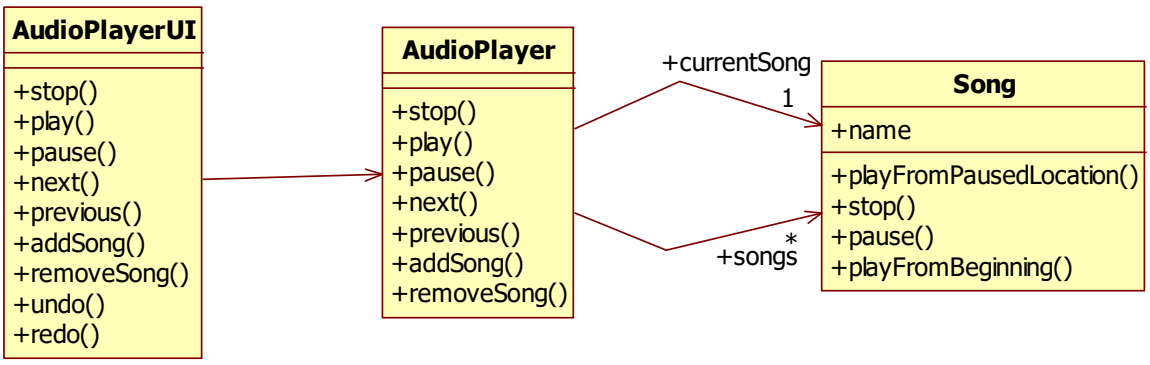
b. Adapter pattern: Modify the code from part a (proxy pattern) so that the client is not coupled to the domain objects. So the service methods getAccount and getAccounts should not return the domain class Account, but an AccountDTO. The adapter should do the conversion between Account and AccountDTO.

c. Suppose you need to design and implement an audio player that has the following user interface:



You can add and remove songs to a list, and the buttons **Stop, Play** and **Pause** operate at the selected song. You can change the selected song with the **Next** and **Previous** buttons.

Your first design looks like this:



This first design has 2 limitations:

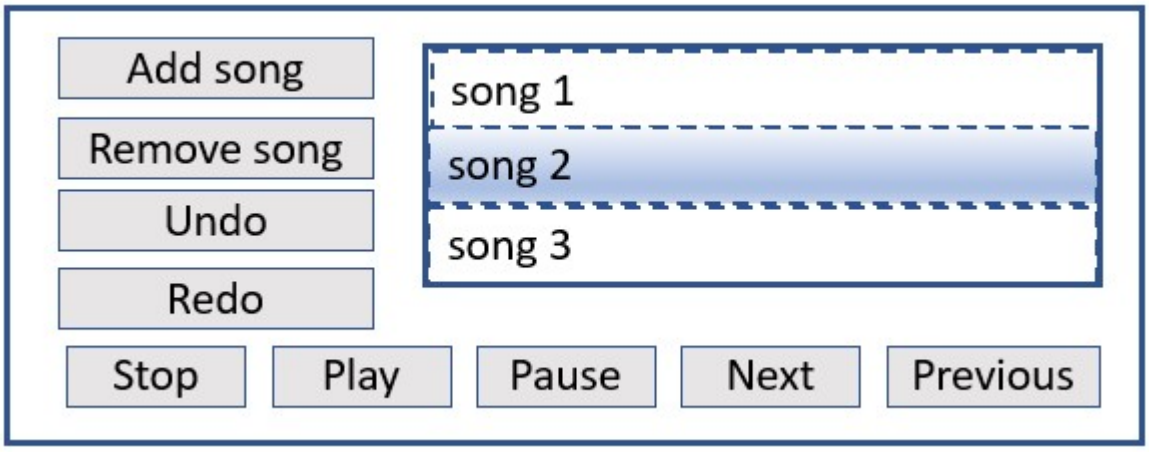
1. The AudioPlayer class contains a lot of conditional logic. For example if we click the Play button, it depends on the current condition of the audio player what will happen. The play() method looks like this:

```
public void play() {
  if (currentState.equals("stop")) {
        currentSong.playFromBeginning();
```

```
            currentState = "play";
        } else if (currentState.equals("pauze")) {
            currentSong.playFromPauzedLocation();
            currentState = "play";
        }
    }
```

We need to get rid of this complex conditional logic.

2. It does not support undo/redo. We have a new requirement that the audio player also needs **undo/redo** support.



a. Draw the class diagram of your design.
b. Draw the **sequence diagram** that shows clearly how your new design works. Suppose the last action that is done is clicking the Stop button. So your sequence diagram starts when the audio player is in the Stop state. Draw the sequence diagram of the following user actions:
    1. The customer clicks the Play button
    2. The user presses the play button again
    3. The customer clicks the Pause button
    4. The customer clicks the Undo button
    5. The customer clicks the Redo button