user.service.ts

```ts
@Injectable ()
export class UserService {
    activeUsers = ['Max', 'Anna'];
    inactiveUsers = ['Chris', 'Manu'];

    constructor(private counter:counterService){}

    setToActive(id:number){
        this.activeUsers.push(this.inactiveUsers[i
        this.inactiveUsers.splice(id,1);
        this.counter.incrementInActiveToActive();
    }

    setToInactive(id:number){
        this.inactiveUsers.push(this.activeUsers[i
        this.activeUsers.splice(id,1);
        this.counter.incrementActiveToInactive();
    }
}
```

Service injected in service

counter.service.ts

```ts
export class counterService{
    actToInact = 0;
    inactToAct = 0;

    incrementActiveToInactive(){
        this.actToInact++;
        console.log('active - inactive : ' + this.actToIna
    }

    incrementInActiveToActive(){
        this.inactToAct++;
        console.log('inactive - active : ' + this.inactTo
    }
}
```

Only One instance of user.service will be given to both the component

active-user.ts

```ts
@Component({
  selector: 'app-active-users',
  templateUrl: './active-users.component.html',
  styleUrls: ['./active-users.component.css']
})
export class ActiveUsersComponent implements  OnI
  users: string[];

  constructor(private userService:UserService){}

  ngOnInit(){
    this.users = this.userService.activeUsers;
  }

  onSetToInactive(id: number){
      this.userService.setToInactive(id);
  }
}
```

inactive-user.ts

```ts
@Component({
  selector: 'app-inactive-users',
  templateUrl: './inactive-users.component.html',
  styleUrls: ['./inactive-users.component.css']
})
export class InactiveUsersComponent implements OnIn
  users: string[];

  constructor(private userService:UserService){}

  ngOnInit(){
    this.users = this.userService.inactiveUsers;
  }

  onSetToActive(id: number) {
    this.userService.setToActive(id);
  }
}
```