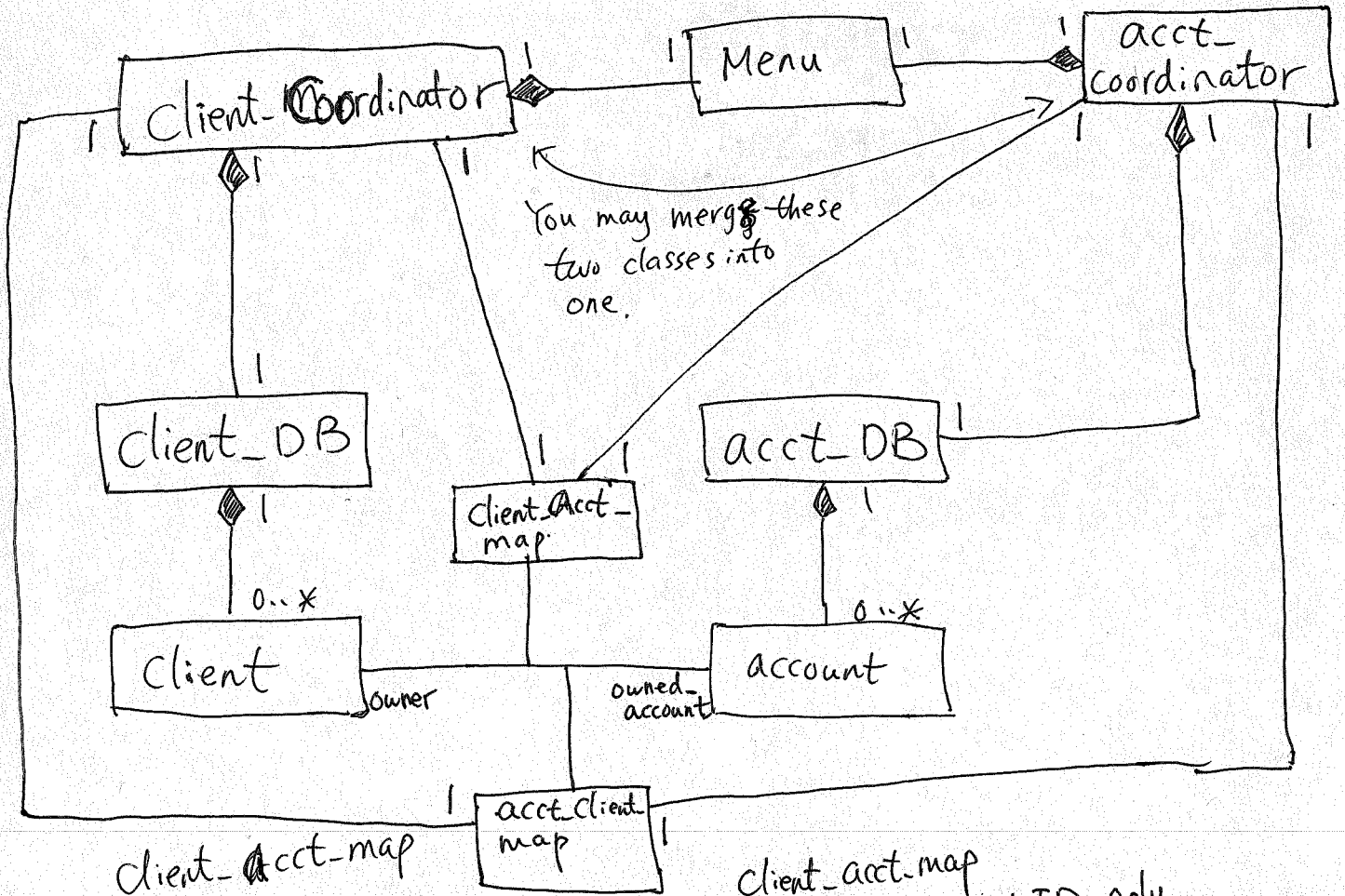


class diagram : { shared accounts support multiple accounts.



client_acct-map

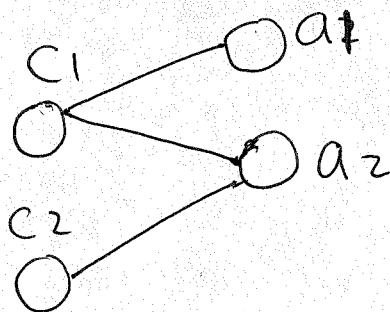
client-ID	acct-ID
C1	a1
C1	a2
C2	a2
⋮	⋮

update to

client_acct-map

C-ID	acct-List
C1	→ [a1 a2]
C2	→ [a1]
⋮	⋮

acct-ID only

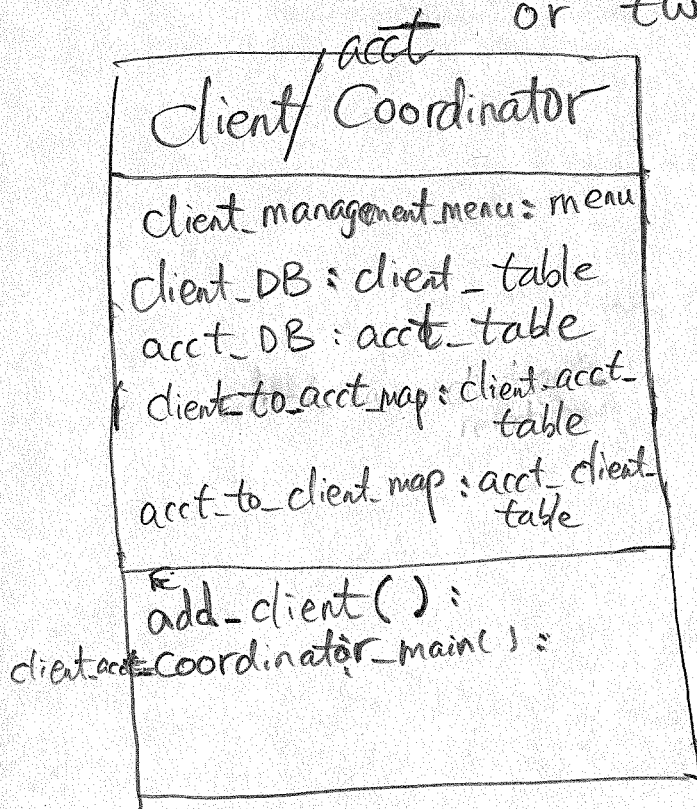


acct_client-map

acct-ID	client-List
a1	→ [C1]
a2	→ [C1 C2]

} client-ID only.

or two coordinators merged into one.



```

client_acct_coordinator :: add_client() {
  get_client_info {
    name
    address
    SSN ← used as ID
    employer
  }
}

```

```

exist ← check_existing_clients(client_to_acct_map, SSN) →
if (exist == Yes) {
  client will not add
else {
  client ← create_a_client(client_info);
  add_client_acct_map(SSN);
  add_acct_client_map(client);
  client_DB ← add_client_DB(client);
}
}
}

```

Do NOT need to add_acct_client_map, because no new acct is created.

```
Client_acct_coordinator := add_acct( ) {
```

```
  Enter name / SSN;
```

```
  exist ← check_client_acct_map(SSN).
```

```
  if (exist == Yes) {
```

```
    Do NOT add;
```

```
  else enter acct-into
```

```
    add_client_acct_map(SSN, acctIDinfo);
```

```
    add_acct_client_map(acct-ID, SSN);
```

```
    create acct object
```

```
    add_acct_DB(acct);
```

```
}
```