Design Splitwise.

⇒ Requirements

→ Add friend Capability

→ Add/Manage Group

→ Add/Manage friends inside a group

→ Manage Expense Inside a group/ without a group

→ Split Expense Capability

- ↳ Equal
- ↳ Unequal
- ↳ Percentage wise

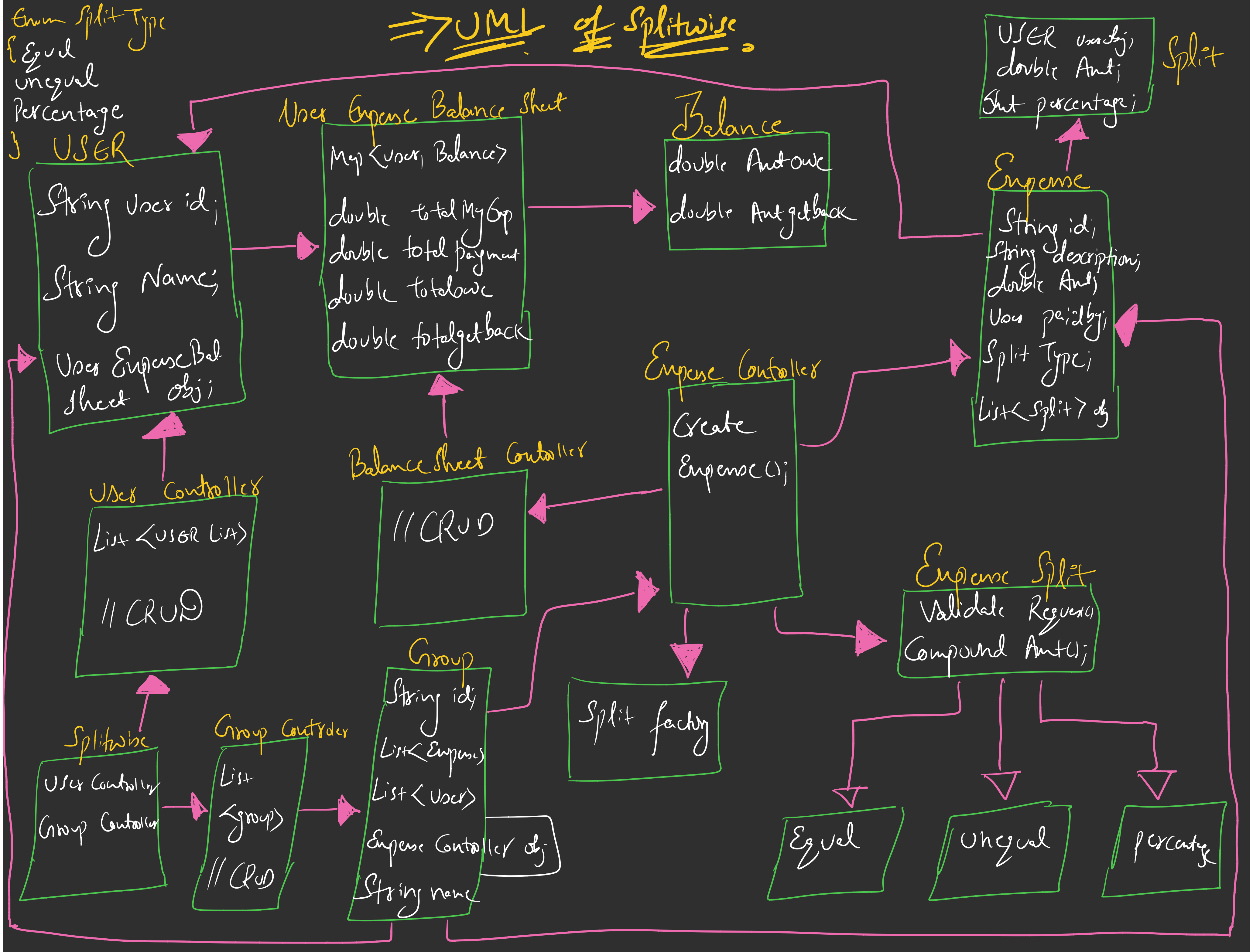
→ Balance Sheet of Each User.

⇒ object Identification

- → USER
- → Group
- → Splitwise
- → Expense
- → Split ???
- → Balance Sheet

[UML Diagram Next Page]

⇒ UML of Splitwise.


Enum Split Type
{ Equal
unequal
Percentage
}
USER
String User id;
String Name;
User ExpenseBal Sheet obj;
User Expense Balance Sheet
Map<User, Balance>
double totalMyExp
double total payment
double totalowe
double totalgetback
Balance
double Amtowe
double Amtgetback
Split
USER userobj;
double Amt;
Int percentage;
Expense
String id;
String description;
double Amt;
User paidby;
Split Type;
List<Split> obj
Expense Controller
Create Expense();
Balance Sheet Controller
//CRUD
User Controller
List<USER List>
//CRUD
Expense Split
Validate Request()
Compound Amt();
Group
String id;
List<Expenses>
List<Users>
Expense Controller obj
String name
Split factory
Splitwise
User Controller
Group Controller
Group Controller
List <groups>
//CRUD
Equal
unequal
percentage