Design Splitwise

Abhitej Voruganti | June 3, 2023

SplitStrategyManager

-splitStrgMgrUnqInst

+getSplitStrategy(SplitFormat): SplitStrategy
+getUnqInst(): SplitStrategyManager

Creates →

<<SplitStrategy>>

+distributeExpense(ExpenseMetaData): void
<<default>>
+updateLedger(User payer, User lender, double amount): void

SplitManager

-splitMgrUnqInst

+distributeExpense(ExpenseMetaData): void
+getUnqInst(): SplitManager

PercentSplitStrategy

+distributeExpense(ExpenseMetaData): void
-isValid(double expense, List<double> expenseBreakUp): bool

EqualSplitStrategy

+distributeExpense(ExpenseMetaData): void
-isValid(List<double> expenseBreakUp): bool

ExactSplitStrategy

+distributeExpense(ExpenseMetaData): void

UserManager

-Map<int userID, User>
-userMgrUnqInst

+show(): void
+show(String userId): void
+addUser(User): void
+getUnqInst(): UserManager

User

-String userId
-String name
-String emailId
-long phoneNum
-Ledger ledger

+shareExpense(): void
+updateLedger(User, double amount): void
+displayFullLedger(): bool
+displayLiabilitiesLedger(): bool

Ledger

-Map<User, double> userBalanceMap

+updateLedger(User, double amount): void
+displayFullLedger(String UserId): bool
+displayLiabilities(String UserId): bool

**<<enumeration>>
SplitFormat**

Equal,
Exact,
Percentage

ExpenseMetaData

-User payer
-List<User> splitAmongUser
-SplitFormat
-int numOfUsers
-List<double> expenseBreakUp
-double expense