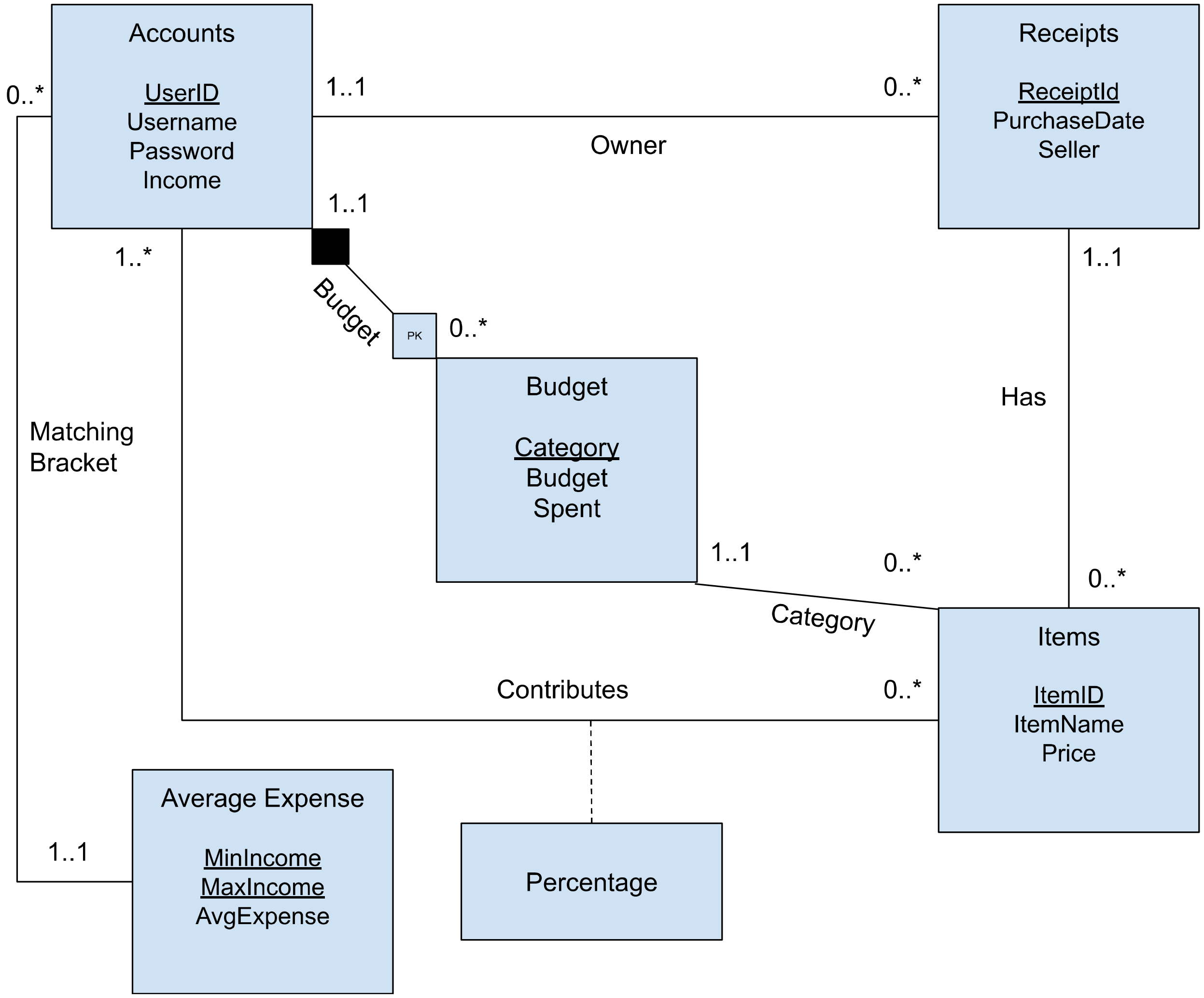
UML Diagram


Accounts
UserID
Username
Password
Income
Receipts
ReceiptId
PurchaseDate
Seller
1..1
0..*
Owner
0..*
1..1
1..*
1..1
Budget
PK
0..*
Budget
Category
Budget
Spent
Has
Matching
Bracket
1..1
0..*
0..*
Category
Items
ItemID
ItemName
Price
Contributes
0..*
Average Expense
MinIncome
MaxIncome
AvgExpense
1..1
Percentage

Assumptions

For each entity

- Accounts (Entity)
    - We need to store each user as a separate entity because login information is different per person.
- Receipts (Entity)
    - We need to uniquely identify each receipt as the items you buy and the amount you pay varies per receipt.
- Items (Entity)
    - Each item needs to be uniquely identified because they have different names and prices.
- Average Expense (Entity)
    - Is an external datasource containing information on the average expense of various categories for different income brackets that is independent of any other information we have
- Budget (Weak Entity)
    - There will be multiple categories and each can have its own budget. Needs to be a weak entity as the budget differs per user and not for the entire database.

For each relation

- Accounts - Receipts (Exactly one to many)
    - Since the user will be uploading receipts to the website, each user can have multiple receipts but each receipt can only belong to exactly one user.
- Accounts - Average Expense (Many to Exactly one)
    - Each user belongs in exactly one income bracket but each income bracket and represent many users
- Accounts - Budget (Exactly one to Many)
    - Each Budget belongs to exactly one user, each user can have budgets for multiple different categories.
- Accounts - Items (Many to Many)
    - Each account can contribute to multiple different items. Each item can be split among multiple accounts.
- Budget - Items (Exactly one to Many)
    - Each Item belongs to exactly the budget category while each budget category can represent multiple items.
- Receipts - Items (Exactly one to Many)
    - Exactly one to many relationships as a receipt can have multiple items and each item on the receipt only belongs to that receipt.

Normalization (3NF)

All of our tables are normalized in 3NF as all attributes are solely dependent on the candidate keys.

Accounts
UserID -> Username, Password, Income, MinIncome, MaxIncome

Receipts
ReceiptId -> UserID, PurchaseDate, Seller

Items
ItemID -> Category, ReceiptID, ItemName

AverageExpense
MinIncome,MaxIncome -> AvgExpense

Budget
Category, UserID -> Budget, Spent

Contributes
UserID, ItemID -> Percentage

Relational Schema

- Accounts(UserID:INT [PK], Username:VARCHAR(255), Password:VARCHAR(255), Income:FLOAT, MinIncome:FLOAT[FK to AverageExpense.MinIncome], MaxIncome:FLOAT[FK to AverageExpense.MaxIncome])

- Receipts(ReceiptID:INT [PK], UserID:INT [FK to Accounts.UserID], PurchaseDate:DATE, Seller:VARCHAR(255))

- Items(ItemID:INT[PK], Category:VARCHAR(255) [FK to Budget.Category], ReceiptID:INT[PK][FK to Receipts.ReceiptID], ItemName:VARCHAR(255), Price:FLOAT)

- AverageExpense(MinIncome:FLOAT[PK], MaxIncome:FLOAT[PK], AvgExpense:FLOAT)

- Budget(Category:VARCHAR(255)[PK], UserID:INT[PK][FK to Accounts.UserID], Budget:FLOAT, Spent:FLOAT)

- Contributes(UserID:INT[PK][FK to Accounts.UserID], ItemID:INT[PK][FK to Items.ItemID], Percentage:FLOAT)