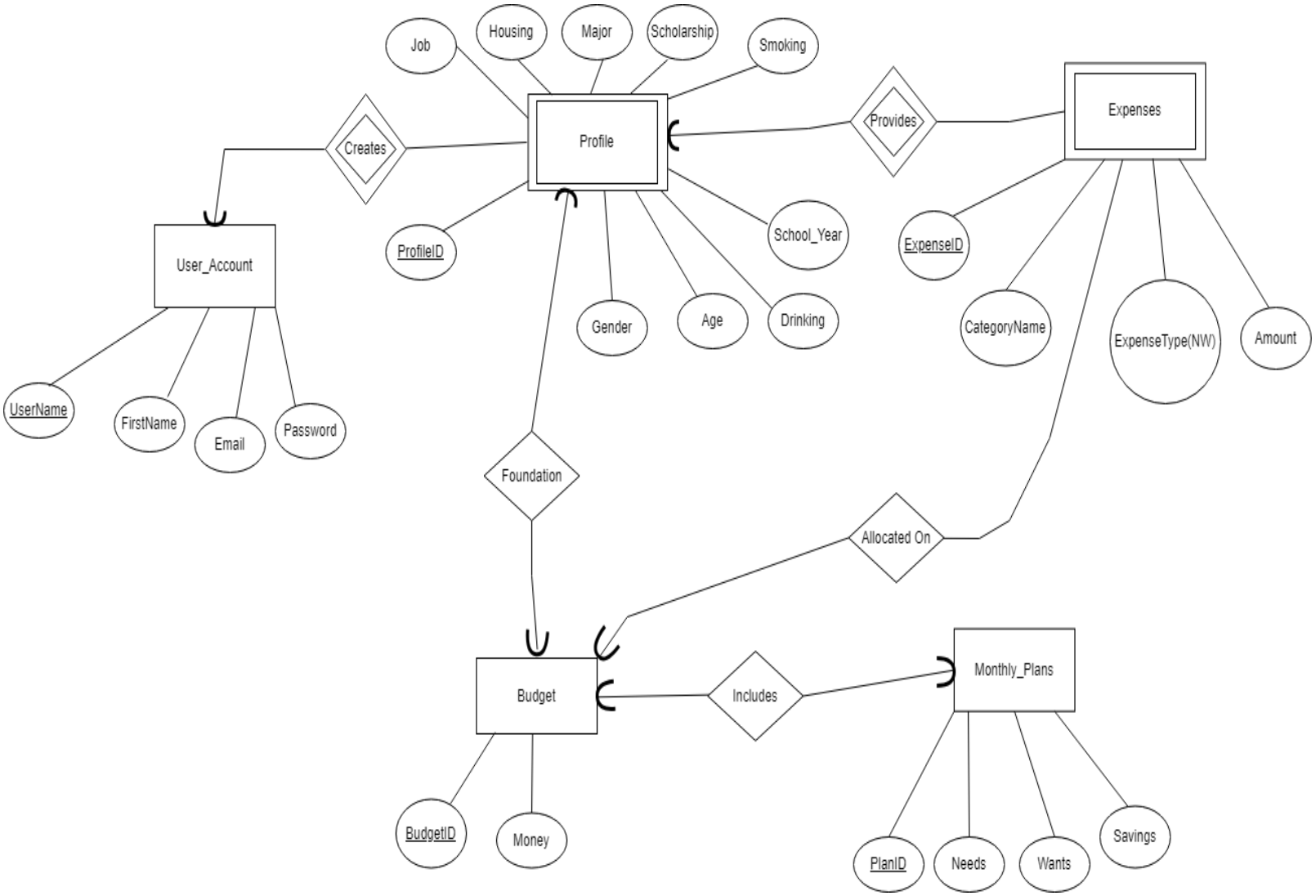
ER Diagram:


Job
Housing
Major
Scholarship
Smoking
Profile
Creates
Provides
Expenses
User_Account
ProfileID
School_Year
ExpenseID
Gender
Age
Drinking
CategoryName
ExpenseType(NW)
Amount
UserName
FirstName
Email
Password
Foundation
Allocated On
Budget
Includes
Monthly_Plans
BudgetID
Money
PlanID
Needs
Wants
Savings

<u>In regards to the suggestions from the previous stages, we will use auto-generated data since the datasets we have most likely will need more records to have 1000 records each for 3 tables.</u>

Database Schema Normalization:
The database schema we have created is normalized since each entity has no other functional dependencies except the one that exists from the primary key of each entity to the other attributes within the same entity. As a result, all of our entities are in normal (3NF) form.

Relational Schema:

User_Account Table

1.) User_Account(UserName VARCHAR(225) [PK], FirstName VARCHAR(225), Email VARCHAR(225), Password VARCHAR()225)

2.) User_Account is an entity because it combines all the various users and the attributes that make each user unique. It is not an attribute because the user/account of the project cannot be fully described in one value.

3.) The User_Account table is in normal form since UserName is a primary key and no other functional dependencies exist within this table except for UserName -> FirstName, Email, and Password. As for the assumptions, the User_Account table will have a singular relationship which is a one-to-many relationship with the Profiles entity since each user will be able to create multiple profiles depending on any changes about the user (i.e. changes to their school year, major, age, job status, personal habits, etc.). Therefore, it makes sense that the User_Account table is a strong entity since it is not dependent on other entities for User_Account to exist. Additionally, the User_Account table is its own entity and not a part of the Profiles entity because of the one to many relationship it has with the Profiles entity.

ProfileCreation Table

1. Relational Schema:
   ProfileCreation(ProfileID VARCHAR(30) [PK] NOT NULL, Gender VARCHAR(10) NOT NULL, Age INT NOT NULL, Job BOOLEAN DEFAULT FALSE, Drinking BOOLEAN DEFAULT FALSE, Housing BOOLEAN DEFAULT FALSE, Major VARCHAR(255), Scholarship BOOLEAN DEFAULT FALSE, Smoking BOOLEAN DEFAULT FALSE, School_Year INT, UserName VARCHAR(225) [FK to User_Account.UserName])

2. Profile needs to be an entity rather than an attribute of another entity because we need the user to create their profile and set each attribute appropriately. In regards to the assumptions, every user will have the ability to create multiple profiles, a many to one relationship (N:1) between Profile and User_Account, and every user profile will have many expense records(1:N). This is because we want to allow users to have multiple Profiles within the same user account based on significant changes about each user during the user's time as a college student. As a result, this makes Profile a weak entity since it cannot exist without a user. Additionally, they should be allowed to keep track of as many expense records as they would like within their account. The Profile entity also has a one to one relationship with the Budget entity since each profile will select a budget that is the most optimal for them.

3. The ProfileCreation relational schema is already in 3NF since ProfileID → is the primary key that determines all other non-key attributes/columns like 'Gender', 'Age', 'Job', etc. And there are no other functional dependencies.

ExpensesProvided Table

1.) Expenses(ExpenseID INT [PK], CategoryName VARCHAR(50), ExpenseType CHAR(3), Amount INT, ProfileID VARCHAR(30) [FK to ProfileCreation.ProfileID], BudgetID VARCHAR(30) [FK to Budget.BudgetID])

2.) Expenses is an entity because the entity is unique and has a column that can identify each instance of an expense. Each expense has multiple attributes that cannot simply be tracked by a single attribute therefore we must have the expenses table as an entity. Likewise, Expenses has multiple relationships between Budget and Profile which would require ExpensesProvided to be a table in order not to show these relationships accurately.

3.) The ExpensesProvided table is in normal form because ExpenseID is a primary key and no other functional dependencies exist within this table except for ExpenseID -> CategoryName, ExpenseType(NW), and Amount. This makes the ExpensesProvided table already in normal (3NF) form. With respect to the assumptions, the ExpensesProvided table will have a many-to-one relationship with the ProfileCreation table because each profile can have many expenses, but each expense belongs to one profile. This makes Expenses a weak entity that depends on the existence of a profile for the Expenses entity to function properly. The same will be true between the expense table and the Budget table as a single BudgetID will be used for multiple expenses. Each user can have multiple expenses but only a single budget, but an expense is not dependent on any budget. The assumption is that each profile will have multiple purchases that will also be linked to their budget. Essentially every expense will take away from their budget.

Monthly_Plans Table

1.) Monthly_Plans(PlanID INT [PK], Needs REAL, Wants REAL, Savings REAL, Budget_ID INT [FK to Budgets.BudgetID])

2.) Monthly_Plans is an entity because it consists of all the different monthly plans for each budget that is made by each user. It holds all the different amounts of money that is separated based on different attributes (Need, Want, and Savings) for each budget that a user will have. These different attributes make sure that the Monthly_Plans table cannot be fully described only through one value. And given that the Monthly_Plans has a one to one relationship with the Budgets table, Monthly_Plans being an entity makes the most sense.

3.) As for the Monthly_Plans table being in normal form, since a PlansID primary key exists and there are no other functional dependencies, Monthly_Plans is in normal (3NF) form. The assumption of Monthly_Plans is that Monthly_Plans has a one to one relationship with the Budgets table. This is because we expect each user to be able to choose one monthly plan out of all the different plans available to them as a preset of monthly plans as they plan their finances, making the Monthly_Plans table a strong entity that is not dependent on Budget and can exist on its own. As a result, it would make most sense for Monthly_Plans and Budgets to have a one to one relationship but not a weak entity relationship.

Budget Table

1.) Budget(BudgetID INT [PK], Money DECIMAL(10, 2) NOT NULL, ProfileID VARCHAR(30) [FK to ProfileCreation.ProfileID])

2.) Budget is an entity because each user will create a distinct budget that will represent how much money they have to use each month. This cannot be represented as an attribute because it needs multiple values such as the amount of money itself and unique identifiers to match it to the user.
3.) The Budget table is in normal (3NF) form because BudgetID is the primary key that determines the other attributes (Money, ProfileID), and there are no other functional dependencies. In terms of assumptions, the Budget table will have a one to one relationship with the Montly_Plans table because the detailed monthly plans that a user chooses will be linked to the budget out of the available option chosen by each user that best fits the user's budget. The Budget table will also have a one to one relationship with the ProfileCreation table because each user can only have one budget, where the user chooses a budget out of the preset budget options available. Because of this, the Budget table is a strong entity that is not dependent on the existence of a profile to exist.

Additionally, the Budget table will have a one to many relationship with the Expenses table because a user's unique BudgetID can be linked to multiple expenses, and these will be subtracted from their total budget.