The Users entity represents the user accounts and user information. Its attributes are UserID, Password, Address, and PhoneNumber. The UserID is the primary key and it is the user email that is unique to the user. This means UserID is type VARCHAR(255). Password is VARCHAR(255), Address is VARCHAR(255), and PhoneNumber is VARCHAR(12) (e.g.123-456-7890). This table is its own entity to ensure that we maintain unique data for each user, as well as to reduce redundancy in the many-to-many relation with the Portfolios table.

Users:	
UserID	VARCHAR(255)
Password	VARCHAR(255)
Address	VARCHAR(255)
PhoneNumber	VARCHAR(12)

Users(UserID:VARCHAR(255) [PK], Password:VARCHAR(255), Address:VARCHAR(255), PhoneNumber:VARCHAR(12))

Portfolios is an entity that will consist of PortfolioID, PortfolioType, and PortfolioBalance. PortfolioID is the primary key and is type INT. Each portfolio will have its own unique identifier. PortfolioType is type VARCHAR(255) and is used to give a short description of the type of portfolio account or even just a name. PortfolioBalance is type DECIMAL(14,2) and keeps track of the cash value of that specific portfolio. Portfolio cannot be an attribute of Users as a user can have multiple portfolios so there would be multiple tuples with almost the same information except the one column. Thus, to remove the redundancy, Portfolios was made an entity. This is to replicate the real-life scenario where a person can open multiple accounts for various reasons such as a Roth, retirement, or individual account.

Portfolios:	
PortfolioID	INT

PortfolioType	VARCHAR(255)
PortfolioBalance	DECIMAL(14,2)

Portfolios(PortfolioID:INT [PK], PortfolioType:VARCHAR(255), PortfolioBalance:DECIMAL(14,2))

Users and Portfolio are connected using a junction table, UsersPortfolios, as it is a many-to-many relationship. A user can have multiple portfolios and a portfolio can have multiple users that own it, making this a many-to-many relation. It would be a portfolio that multiple users collaborate on. A junction allows for easier linking between the two entities without having to overfill the entities with much redundant information. Its attributes are UserID and PortfolioID, which are the composite primary keys. Individually, they are foreign keys: UserID references Users and PortfolioID references Portfolios.

UsersPortfolios	
UserID FK to Users	VARCHAR(255)
PortfolioID FK to Portfolios	INT

UsersPortfolios(UserID:VARCHAR(255) [PK] [FK to User.UserID], PortfolioID:INT [PK] [FK to Portfolios.PortfolioID])

Transactions is an entity that consists of all transaction information. Its attributes are TransactionID, PortfolioID, CurrentlyActive, TransactionType, StockSymbol, NumShares, PurchasePrice, DateTime. TransactionID is the primary key with type INT. PortfolioID is a foreign key that references the Portfolios table, where PortfolioID is type INT. CurrentlyActive is type BOOLEAN and indicates whether the stock is currently active in the linked portfolio or whether it has been sold. TransactionType is type VARCHAR(4), constrained to either SELL or BUY, indicating the type of transaction that occurred. StockSymbol is type VARCHAR(10) and symbolizes which company stock was bought/sold from. NumShare is type INT, indicating the number of shares the user

purchased/sold. It is type INT for the purpose of keeping this site as a simple introduction for people just starting out with investing. PurchasePrice is type DECIMAL(14,2) and indicates the price of the stock when the transaction occurs. This rounds off the value to the nearest hundredth. This information is collected through an API. DateTime is type TIMESTAMP and records when the transaction occurred as a log of the transaction. Transactions is an entity because each portfolio can have numerous transactions, so it would be redundant to have it as an attribute of Portfolios. Also, Transactions has its own attributes that we need to keep a record of. There are multiple transactions linked to a portfolio because the user can buy and sell multiple stocks however many times they want. A Transaction is linked to only one specific Portfolio as users can only buy/sell from within a specific portfolio, but a portfolio can have multiple transactions.

Transactions	
TransactionID	INT
PortfolioID FK to Portfolios	INT
Currently Active	BOOL
TransactionType	VARCHAR(4)
StockSymbol	VARCHAR(10)
NumShares	INT
PurchasePrice	DECIMAL(14,2)
DateTime	TIMESTAMP

Transactions(TransactionID:INT [PK]: PortfolioID:INT [FK to Portfolios.PortfolioID], CurrentlyActive:BOOLEAN, TransactionType:VARCHAR(4 StockSymbol:VARCHAR(10), NumShares:INT, PurchasePrice:DECIMAL(14,2), DateTime:TIMESTAMP)

WatchList is an entity that keeps track of certain stocks in each portfolio that the users want quick access to. The attributes are StockSymbol and PortfolioID. Both are the primary keys and PortfolioID is a foreign key: PortfolioID, type INT, references

Portfolios. StockSymbol, type VARCHAR(10), is a primary key because, in each portfolio, the watchlist will list only one of the stock information, so combining it with the PortfolioID will make it a unique combination. This watchlist will be limited to a specific quantity and act as “user favorited”. The reason there aren’t other attributes is that the information, such as the price of the stock, will be collected when the user clicks on a specific portfolio using an API, and since it will be collected in real time, it won’t need to be inserted into the database. A portfolio can have 0-n stocks in its watchlist but each watchlist item has only one portfolio it is linked to, making it a 0-to-many relation. The reason WatchList is only linked to one portfolio is that the user would have different stocks to watch for different portfolios, as the portfolios may have been made for different reasons. So each portfolio would have its own multitude of stocks to watch. There are likely going to be multiple WatchList items for each portfolio so it would be more efficient to store these as a separate table to avoid redundancy.

WatchList	
StockSymbol	VARCHAR(10)
PortfolioID FK to Portfolios	INT

WatchList(StockSymbol:VARCHAR(10) [PK], PortfolioID:INT [PK] [FK to Portfolios.PortfolioID])

Logins is an entity that keeps a log of user sign-ins. The attributes are UserID and DateTime. UserID is a primary key that links the login record to a user, it is also a foreign key from the Users table. DateTime, type TIMESTAMP, is also a primary key in combination with UserID as a user can log in multiple times, so this composite key will allow for a unique identifier. This table is used as a security feature to keep track of user log-ins. A user can have multiple logins but a login can only have one user, making a

one-to-many relation. We kept this as a separate entity as there are bound to be multiple logins for each user so it is more efficient to store this data in a separate table.

LogIns	
UserID FK to Users	VARCHAR(255)
DateTime	TIMESTAMP

Logins(UserID:VARCHAR(255) [PK] [FK to Users.UserID], DateTime:DATETIME [PK])

Cardinality Explanations:

Users <--> Portfolios:

Multiple Users can collaborate on one Portfolio, and one User can have multiple Portfolios, making this a many-to-many relationship.

Users <--> Logins:

A user can have zero to many logins or logouts, but a login can only have one user, constituting a one-to-many relation.

Portfolios <--> Watchlist:

A portfolio will have 0-(some max number) watchlist items, and each watchlist is assigned to only one portfolio, making this a one-to-many relation.

Portfolios <--> Transaction:

A portfolio can have zero to many transactions but each transaction is linked to only one portfolio, making this a one-to-many relation.