

# CSCC43 Project

CSCC43

Tom Kim 1009502940

Vincent Wang 1009411557

August 2024

# 1 Entity Relationship Diagram

The entity relationship diagram (ERD) is displayed in figure 1 on page 10.

## 2 Relational Schema

The following relational schema is derived based on the ERD developed above.

- Entity sets:

Accounts(userID, firstName, lastName, email, password)

StockLists(stockListID, stockListName, category)

Reviews(userID, stockListID, timestamp, response)

Portfolios(portfolioID, portfolioName, cashAvailable)

Stocks(symbol, stockName)

DailyStocks(symbol, timestamp, open, close, high, low, volume)

- Relationships:

ReviewRequests(sent, received, stockListID, dateTime)

Friends(user1, user2, status, dateTimeAddedDeleted)

FriendRequests(sent, received, status, dateTime)

OwnsPortfolio(userID, portfolioID)

OwnsStockList(userID, stockListID)

AccessibleBy(userID, stockListID)

StockListStocks(stockListID, symbol, numShares)

PortfolioStocks(portfolioID, stockListID)

### 3 Normalization

Decomposition into BCNF will be completed to remove as much redundancy of information as possible and to remove update, insertion, and deletion anomalies.

- Accounts(userID, firstName, lastName, email, password)

The FD(s) are:

$\text{userID} \rightarrow \text{firstName}, \text{lastName}, \text{email}, \text{password}$  (satisfies BCNF since userID is a superkey)

$\text{email} \rightarrow \text{userID}, \text{firstName}, \text{lastName}, \text{password}$  (satisfies BCNF since email is a superkey – it determines all other attributes)

- StockLists(stockListID, stockListName, category)

The FD(s) are:

$\text{stockListID} \rightarrow \text{stockListName}, \text{category}$  (satisfies BCNF since stockListID is a superkey)

- Reviews(userID, stockListID, timestamp, response)

The FD(s) are:

$\text{userID}, \text{stockListID} \rightarrow \text{timestamp}, \text{response}$  (satisfies BCNF since userID, stockListID is a superkey)

- Portfolios(portfolioID, portfolioName, cashAvailable)

The FD(s) are:

$\text{portfolioID} \rightarrow \text{portfolioName}, \text{cashAvailable}$  (satisfies BCNF since portfolioID is a superkey)

- Stocks(symbol, stockName)

The FD(s) are:

$\text{symbol} \rightarrow \text{stockName}$  (satisfies BCNF since symbol is a superkey)

- DailyStocks(symbol, timestamp, open, close, high, low, volume)

The FD(s) are:

symbol, timestamp  $\rightarrow$  open, close, high, low, volume (satisfies BCNF since symbol, timestamp is a superkey)

- ReviewRequests(sent, received, stockListID, dateTime)

The FD(s) are:

sent, received, stockListID  $\rightarrow$  dateTime (satisfies BCNF since sent, received, stockListID is a superkey)

- Friends(user1, user2, status, dateTimeAddedDeleted)

The FD(s) are:

user1, user2  $\rightarrow$  status, dateTimeAddedDeleted (satisfies BCNF since user1, user2 is a superkey)

- FriendRequests(sent, received, status, dateTime)

The FD(s) are:

sent, received  $\rightarrow$  status, dateTime (satisfies BCNF since sent, received is a superkey)

- OwnsPortfolio(userID, portfolioID)

The FD(s) are:

portfolioID  $\rightarrow$  userID (satisfies BCNF since portfolioID is a superkey)

- OwnsStockList(userID, stockListID)

The FD(s) are:

stockListID  $\rightarrow$  userID (satisfies BCNF since stockListID is a superkey)

- AccessibleBy(userID, stockListID)

There are no FDs for this relation.

- StockListStocks(stockListID, symbol, numShares)

The FD(s) are:

$\text{stockListID, symbol} \rightarrow \text{numShares}$  (satisfies BCNF since stockListID, symbol is a superkey)

- PortfolioStocks(portfolioID, stockListID)

The FD(s) are:

$\text{portfolioID} \rightarrow \text{stockListID}$  (satisfies BCNF since portfolioID is a superkey)

All relations are already in BCNF. Therefore, no further decomposition is necessary.

## 4 SQL Implementation

Please check the attached file to see all the SQL used in the code, organized by functionality.

### 4.1 Relation Creation

Example of SQL used in relation creation:

```
CREATE TABLE IF NOT EXISTS Accounts(  
    userID SERIAL PRIMARY KEY,  
    firstName VARCHAR(30),  
    lastName VARCHAR(30),  
    email VARCHAR(100) CHECK(email LIKE '_%@_%._%') UNIQUE,  
    password VARCHAR(100),  
);
```

### 4.2 Review Functionality

Example of SQL used in the review functionality:

```
INSERT INTO Reviews(userID, stockListID, response)  
VALUES(<loggedInUserID>, <stockListID>, <text>)
```

### 4.3 Portfolio Functionality

Example of SQL used in the portfolio functionality:

```
SELECT portfolioID, portfolioName, cashAvailable
FROM OwnsPortfolio NATURAL JOIN Portfolios
WHERE userID = <loggedInUserID>
```

### 4.4 Stock Functionality

Example of SQL used in the stock functionality:

```
SELECT timestamp, open, close, high, low, volume
FROM DailyStocks
WHERE symbol = '<stockSymbol>'
      AND timestamp >= '<startTime>'
      AND timestamp <= '<endTime>'
ORDER BY timestamp ASC
```

### 4.5 Stock List Functionality

Example of SQL used in the stock list functionality:

```
DELETE FROM StockLists WHERE stockListID = <stockListID>
```

### 4.6 Account Functionality

Example of SQL used in the account functionality:

```
INSERT INTO Accounts(firstName, lastName, email, password)
VALUES('<firstName>', '<lastName>', '<email>', '<password>')
RETURNING userID
```

## 4.7 Friend Functionality

Example of SQL used in the friend functionality:

```
(SELECT user1 AS friend, firstName, lastName, email
FROM Accounts JOIN Friends ON Accounts.userID = Friends.user1
WHERE user2 = '<loggedInUserID>' AND status = 'added')

UNION

(SELECT user2 AS friend, firstName, lastName, email
FROM Accounts JOIN Friends ON Accounts.userID = Friends.user2
WHERE user1 = '<loggedInUserID>' AND status = 'added')
```

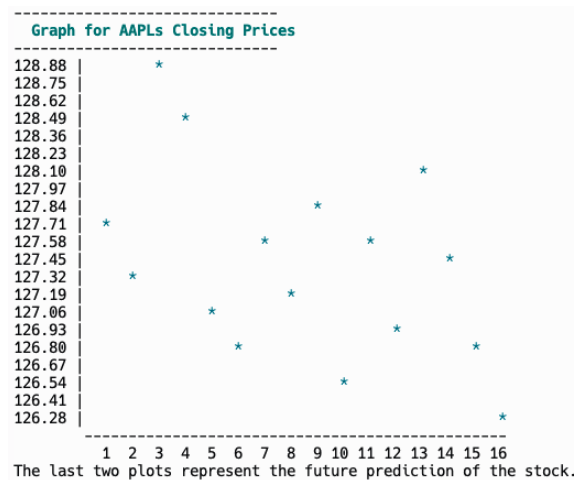
## 5 Appendix

### 5.1 Extra Credit Request

We would appreciate extra credit towards the project grade for the optimized interface.

The interface is intuitive to use and has optimized visuals.

The plot graphs are easy to read with colours used in the plots and the axes are separated from the plots with lines.



Menus have a title with bold colour text. Options available are neatly placed next to their corresponding number that the user can select. Prompts are in a distinct colour so the user is easily able to recognize that they can input their response.

```

Options
-----
1 | Send friend request
2 | View friend requests
3 | View friends
4 | Create a new portfolio
5 | View your portfolios
6 | Create a new stock list
7 | View your stock lists
8 | View stock lists from other users
9 | Add stock data
10 | View review requests
11 | Log out
12 | Exit
-----
Enter your choice:
1
Enter the email address of the friend you would like to add:
p@p.com

```

Tables have a title with bold colour text. Lines are used to separate the title, column headers, the data, and each individual column.



Information for AAPL							
Number	Date	Open	Close	High	Low	Volume	
1	2014-05-05	84.3056	85.8513	85.8571	84.2856	71766758	
2	2014-05-06	85.9713	84.9156	86.3442	84.9156	93641373	
3	2014-05-07	85.0356	84.6185	85.3271	83.9613	70715988	
4	2014-05-08	84.0356	83.9985	84.9156	83.7713	57574363	
5	2014-05-09	83.5056	83.6488	83.7499	82.9042	72899498	
6	2014-05-12	83.9271	84.6899	84.8085	83.9142	53324677	
7	2014-05-13	84.5713	84.8228	84.9342	84.3856	39934594	
8	2014-05-14	84.6328	84.8385	85.3428	84.5342	41600846	
9	2014-05-15	84.9571	84.1171	85.2285	84.0056	57711731	
10	2014-05-16	84.0899	85.3585	85.3613	83.6285	69091834	
11	2014-05-19	85.4071	86.3699	86.7613	85.3328	79439024	
12	2014-05-20	86.3585	86.3071	86.6285	85.8185	50700906	
13	2014-05-21	86.2613	86.6156	86.6713	86.0085	49249914	
14	2014-05-22	86.6571	86.7528	87.1213	86.2999	50218945	
15	2014-05-23	86.7499	87.7328	87.8184	86.6385	58052491	
16	2014-05-27	87.9828	89.3756	89.4085	87.9471	87216605	
17	2014-05-28	89.4313	89.1442	89.9756	89.1113	78921885	
18	2014-05-29	89.6928	90.7685	90.9813	89.6813	94118633	
19	2014-05-30	91.1399	90.4285	92.0242	89.8428	141005137	
20	2014-06-02	90.5656	89.8071	90.6899	88.9285	92337903	
21	2014-06-03	89.7799	91.0771	91.2485	89.7499	73231620	
22	2014-06-04	91.0628	92.1171	92.5556	90.8728	83870521	
23	2014-06-05	92.3142	92.4785	92.767	91.8013	75951141	

## 5.2 Entity Relationship Diagram

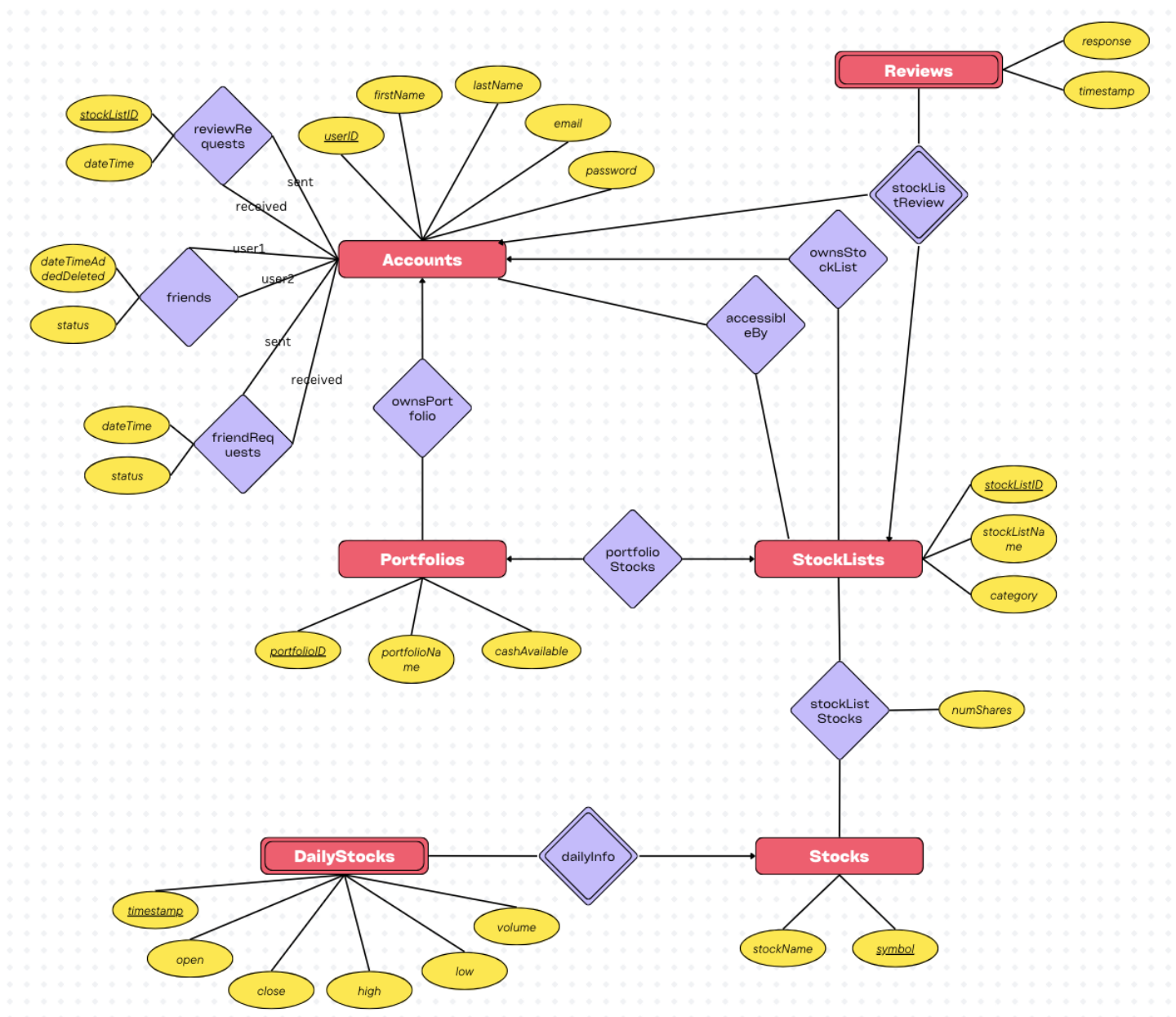


Figure 1: The ERD for the application database.