CPSC 304 Project Cover Page

Milestone #: 2

Date: 10 / 20 / 2023

Group Number: 33

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Brian Zhou	50244482	h4o1o	brianhzhou2004@gmail.com
Mohsen Bakhit	47283742	o7n3k	bakhitmohsen@proton.me
Muneeba Ashiq	79455622	g9v7y	muneeba.ashiq02@gmail.com

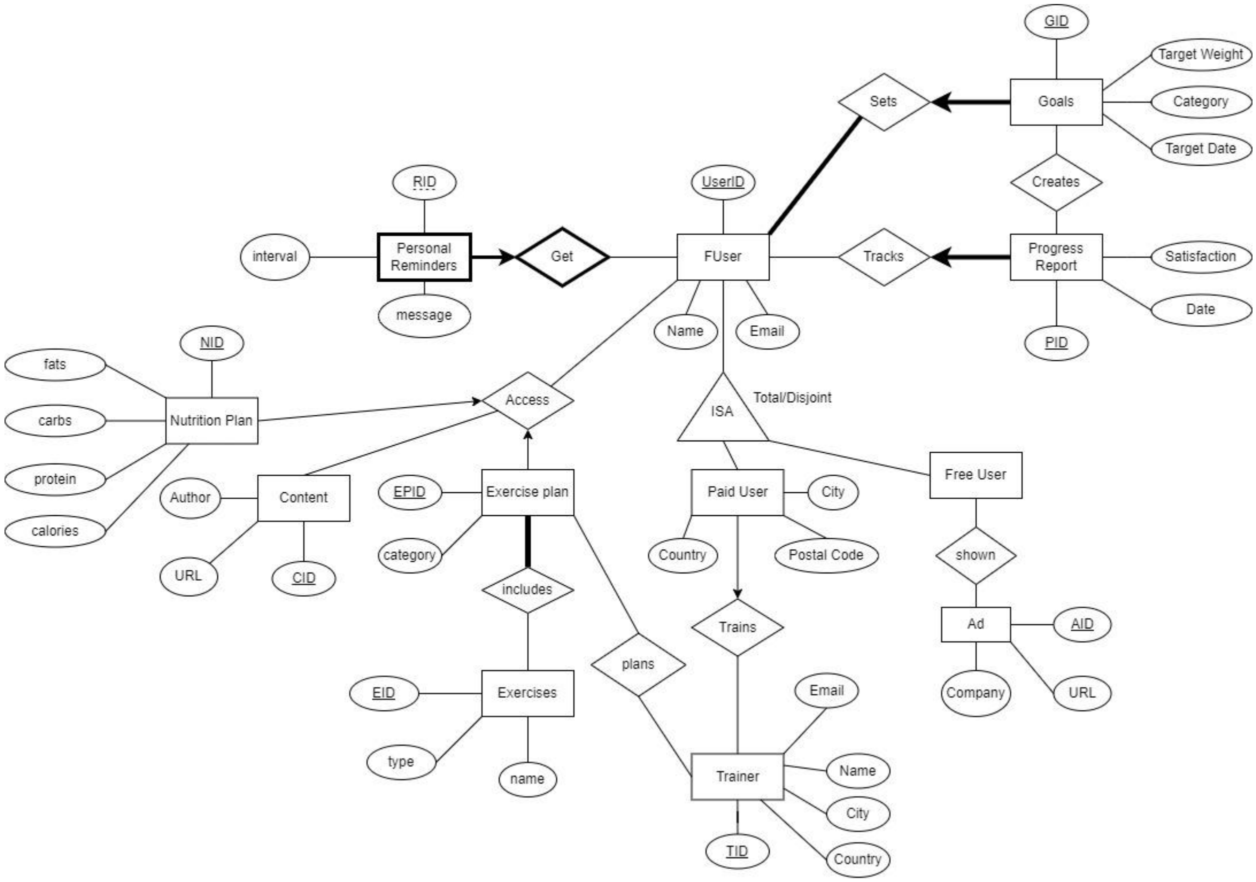
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Summary:

A fitness web app that helps users set goals and track them over time with reports and notifications, creates and gives access to workouts, nutrition plans and articles related to fitness.

ER Diagram:



Added additional attributes to most entities to store more meaningful information.

Add Exercise Plan entity to allow Trainers to compile custom exercises for Paid Users

Following advice, added Ads Entity for Free Users to give them a unique experience!

4. Schema Derived from ER Diagram

Bold means foreign key. Underlined means part of the primary key.

User(userid : int , Name: Char(20), Email: Char(100) UNIQUE)

Paid User (**userid**: int, Postal Code: Char (12), Country: Char (15) NOT NULL, City: Char (20) NOT NULL)

Goals (GID : int, Category : Char (20), TargetDate : Date, TargetWeight : int, **userid**: int)

ProgressReport (PID : int, Satisfaction : Char (20), date: Date, **userid**: int)

Creates(**GID**: int, **PID**: int)

Ad(AdID: int, Company: Char (50), URL: Char(100))

Shown(**userid**: int, **AdID**: int)

Trains(**TID**: int, **userid**: int)

Trainer (TID : int, Name: Char (20), Email: Char(100), City: Char(20), Country: Char(15))

Plans(**TID**: int, **EPID**: int)

Personal Reminders(RID: int, interval: int, **userid**:int, msg: Char (150))

Nutrition Plan (NID : int, Calories: int, Carbs : int, Fats : int, Protein : int, **userid**: int)

Exercise Plan (EPID : int, Type : Char(20), **TID**: int, **userid**: int)

Exercise (EID : int, Type : Char (20), Name : Char (20))

Content (CID : int, URL: Char (100), Author : Char (20))

Includes (**EPID** : int, **EID** : int)

5. Functional Dependencies

User.userid -> FUser.name, FUser.Email

PaidUser.userid -> PaidUser.Country, PaidUser.PostalCode, PaidUser.City
PaidUser.PostalCode, PaidUser.Country -> PaidUser.City
Goals.GID -> Goals.Category, Goals.TargetDate, Goals.TargetWeight, FUser.userid
ProgressReport.PID -> ProgressReport.Satisfaction, FUser.userid
Ad.AdID-> AdId.Company, AdID.URL
Trainer.TID -> Trainer.Email, Trainer.City, Trainer.Country, TrainerName
NutritionPlan.NID -> NutritionPlan.Calories, NutritionPlan.Carbs, NutritionPlan.Fats, NutritionPlan.Protein, FUser.userid
ExercisePlan.EPID -> ExercisePlan.Type, ExercisePlan.TID, ExercisePlan.userid
Exercise.EID -> Exercise.Type, Exercise.Name
PersonalReminder.RID, FUser.userid -> PersonalReminder.interval, PersonalReminder.msg
NutritionPlan.fats, NutritionPlan.protein, NutritionPlan.Carbs -> NutritionPlan.Calories
Content.CID -> Content.URL, Content.Author

6. Normalization

User(userid, Name, Email)

- Already in BCNF

PaidUser (**userid**, Postal Code, Country, City)

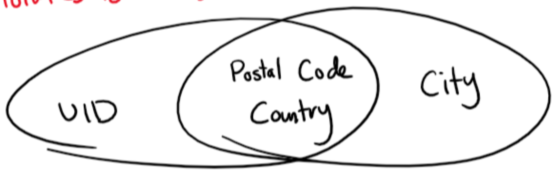
- Not in BCNF, needs to decompose

PaidUser(UID, postal code, country, city)

UID → Country, Postal Code, City

Postal Code, Country → City

Violates BCNF, decompose



$R_1(\underbrace{\underline{UID}}_{\text{primary key}}, \underbrace{\underline{Postal Code}}_{\text{foreign key}}, \underbrace{\underline{Country}}_{\text{foreign key}})$

$R_2(\underbrace{\underline{Postal Code}}_{\text{candidate key}}, \underbrace{\underline{Country}}_{\text{candidate key}}, City)$

-
- PaidUser1(userid, **PostalCode**, **Country**)
    - In BCNF
- PaidUser2(PostalCode, Country, City)
    - In BCNF

Goals (GID, Category, TargetDate, TargetWeight, **userid**)

- Already in BCNF

ProgressReport (PID, Satisfaction, date, **userid**)

- Already in BCNF

Creates(**GID**, **PID**)

- Already in BCNF

Ad(AdID, Company, URL)

- Already in BCNF

Shown(**userid**, **AdID**)

- Already in BCNF

Trains(**TID**, **userid**)

- Already in BCNF

Trainer (TID, Name, Email, City, Country)

- Already in BCNF

Plans(**TID**, **EPID**)

- Already in BCNF

Personal Reminders(RID, interval, msg, **userid**)

- Already in BCNF

Nutrition Plan (NID, Calories, Carbs, Fats, Protein, **userid**)

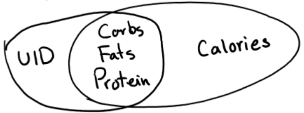
- Not in BCNF, need to decompose

Nutrition Plan (NID, Calories, Carbs, Fats, Protein, UID)

NID→ Calories, Carbs, Fats, Protein, UID

Carbs, Fats, Protein → Calories

violates BCNF, decompose



$R_1(\underset{pk}{\underline{UID}}, \underset{fk}{\underline{fats}}, \underset{fk}{\underline{carbs}}, \underset{fk}{\underline{protein}})$

$R_2(\underset{ck}{\underline{fats}}, \underset{ck}{\underline{carbs}}, \underset{ck}{\underline{protein}}, calories)$

-
- NutritionPlan1(**userid**, **fats**, **carbs**, **protein**)
    - In BCNF
- NutritionPlan2(fats, carbs, protein, calories)
    - In BCNF

Exercise Plan (EPID, Type, **TID**, **userid**)

- Already in BCNF

Exercise (EID, Type, Name)

- Already in BCNF

Content (CID, URL, Author)

- Already in BCNF

Includes (**EPID** , **EID**)

- Already in BCNF

7. SQL DDL STATEMENT

```
CREATE TABLE FUser(
 userid INTEGER,
 name VARCHAR(20),
 email VARCHAR(20) UNIQUE,
 PRIMARY KEY (userid)
);

CREATE TABLE PaidUser2(
postalCode VARCHAR(7),
country VARCHAR(30),
city CHAR(20),
PRIMARY KEY (postalCode, country)
);
```

```
CREATE TABLE PaidUser1(
  userid INTEGER,
  postalCode VARCHAR(7),
  country VARCHAR(30),
  PRIMARY KEY (userid,  postalCode, country),
  FOREIGN KEY (postalCode, country) REFERENCES PaidUser2(postalCode, country),
  FOREIGN KEY (userid) REFERENCES FUser(userid)
);


CREATE TABLE Goals(
   gid INTEGER,
   category VARCHAR(20),
   targetDate DATE,
   targetWeight INTEGER,
   userid INTEGER,
   primary key (gid),
   foreign key (userid) references Fuser(userid)
 );

 CREATE TABLE progressreport(
   pid integer,
   satisfaction integer,
   reportdate date,
   userid integer,
   primary key (pid),
   foreign key (userid) references fuser(userid)
);

 create table creates(
   gid integer,
   pid integer,
   primary key (gid,pid),
   foreign key (gid) references goals(gid),
   foreign key (pid) references progressreport(pid)
);

create table ad(
  adID integer,
  company VARCHAR(50),
  url VARCHAR(100),
  primary key (adID)
);
```

```
CREATE TABLE PersonalReminders(
 rid INTEGER,
 userid INTEGER,
 remindinterval INTEGER,
 msg VARCHAR(150),
 PRIMARY KEY (rid, userid),
 FOREIGN KEY (userid) REFERENCES FUser(userid)
);

create table shown(
  adid integer,
  userid integer,
  primary key(adid, userid),
  foreign key (adid) references ad(adid),
  foreign key (userid) references fuser(userid)
);

create table trainer(
  tid integer,
  name VARCHAR(50),
  email VARCHAR(50),
  city VARCHAR(20),
  country VARCHAR(20),
  primary key (tid)
);

create table trains(
  tid integer,
  userid integer,
  primary key (tid, userid),
  foreign key (tid) references trainer(tid),
  foreign key (userid) references fuser(userid)
);

CREATE TABLE ExercisePlan (
epid INTEGER,
type VARCHAR(20),
tid INTEGER,
userid INTEGER,
PRIMARY KEY (epid)
);

CREATE TABLE Exercise (
eid INTEGER,
type VARCHAR(20),
name VARCHAR (30),
PRIMARY KEY (eid)
);

CREATE TABLE Content (
cid INTEGER,
author VARCHAR(50),
url VARCHAR (100),
PRIMARY KEY (cid)
);


CREATE TABLE Includes (
epid INTEGER,
eid INTEGER,
PRIMARY KEY (epid, eid),
FOREIGN KEY (epid) REFERENCES ExercisePlan(epid),
FOREIGN KEY (eid) REFERENCES Exercise(eid)
);

CREATE TABLE NutritionPlan2(
fats INTEGER,
carbs INTEGER,
protein INTEGER,
calories INTEGER,
PRIMARY KEY (fats, carbs, protein)
);

CREATE TABLE NutritionPlan1(
userid INTEGER,
fats INTEGER,
carbs INTEGER,
protein INTEGER,
PRIMARY KEY (userid),
FOREIGN KEY (userid) REFERENCES FUser(userid),
FOREIGN KEY (fats, carbs, protein) REFERENCES NutritionPlan2(fats,carbs,protein)
);
```

8. SQL INSERT STATEMENT

FUser

```
    INSERT INTO FUser VALUES (1, 'Stanley Ed', 'stanh@mail.com');
    INSERT INTO FUser VALUES (2, 'Peter Bret', 'pbj2003@mail.com');
    INSERT INTO FUser VALUES (3, 'Genghis Khan', 'jkhan@mail.com');
    INSERT INTO FUser VALUES (4, 'Mr. Muffins', 'mrMuff@mail.com');
    INSERT INTO FUser VALUES (5, 'Muhammad Wang', 'mwang@mail.com');
    INSERT INTO FUser VALUES (6, 'Allan Kim', 'ak@hotmail.ca');
    INSERT INTO FUser VALUES (7, 'Ian Hemlock', 'btree@mail.com');
    INSERT INTO FUser VALUES (8, 'Melody Guy', 'mrsguy@hotmail.com');
```

PaidUser1

```
    INSERT INTO PaidUser1 VALUES (1, 'abc def', 'canada');
    INSERT INTO PaidUser1 VALUES (2, 'abc 123', 'usa');
    INSERT INTO PaidUser1 VALUES (3, '123 456', 'china');
    INSERT INTO PaidUser1 VALUES (4, '345 985', 'india');
    INSERT INTO PaidUser1 VALUES (5, 'adf sdf', 'iraq');
```

PaidUser2

```
    INSERT INTO PaidUser2 VALUES ('abc def', 'canada', 'vancouver-ca');
    INSERT INTO PaidUser2 VALUES ('abc 123', 'usa', 'oregon-us');
    INSERT INTO PaidUser2 VALUES ('123 456', 'china', 'shanghai-cn');
    INSERT INTO PaidUser2 VALUES ('345 985', 'india', 'chennai-in');
    INSERT INTO PaidUser2 VALUES ('adf sdf', 'iraq', 'baghdad-iq');
```

Goals

```
INSERT INTO Goals VALUES (1, 'Weight Loss', '10-sep-23', 100, 1);
INSERT INTO Goals VALUES (2, 'Powerlifting', '10-sep-23', 120, 2);
INSERT INTO Goals VALUES (3, 'Weight Loss', '10-sep-23', 80, 3);
INSERT INTO Goals VALUES (5, 'Conditioning', '10-sep-23', 85, 4);
INSERT INTO Goals VALUES (4, 'Bulking Up', '10-sep-23', 140, 5);
```

ProgressReport

```
INSERT INTO ProgressReport VALUES (1, 7, '10-sep-2023', 1);
INSERT INTO ProgressReport VALUES (2, 3, '17-sep-2023', 1);
INSERT INTO ProgressReport VALUES (3, 9, '10-sep-2023', 2);
INSERT INTO ProgressReport VALUES (4, 6, '19-sep-2023', 2);
INSERT INTO ProgressReport VALUES (5, 6, '16-sep-2023', 3);
```

Creates

```
        INSERT INTO Creates VALUES (1, 1);
        INSERT INTO Creates VALUES (2, 2);
        INSERT INTO Creates VALUES (3, 3);
        INSERT INTO Creates VALUES (4, 4);
        INSERT INTO Creates VALUES (5, 5);
```

Ad

```
INSERT INTO Ad VALUES (1, 'Google', 'www.google.com');
INSERT INTO Ad VALUES (2, 'facebook', 'www.facebook.com');
INSERT INTO Ad VALUES (3, 'Farmer Joes', 'www.farmerjoes.com');
INSERT INTO Ad VALUES (4, 'Black Rock', 'www.blackrock.com');
INSERT INTO Ad VALUES(5, 'Home Depot', 'www.homedepot.com');
```

Shown

```
        INSERT INTO Shown VALUES (1, 1);
        INSERT INTO Shown VALUES (1, 2);
        INSERT INTO Shown VALUES (1, 3);
        INSERT INTO Shown VALUES (1, 4);
        INSERT INTO Shown VALUES (2, 1);
```

Trains

```
INSERT INTO Trains VALUES (1,1);
INSERT INTO Trains VALUES (2,4);
INSERT INTO Trains VALUES (4,2);
INSERT INTO Trains VALUES (5,5);
INSERT INTO Trains VALUES (3,3);
```

Trainer

```
        INSERT INTO Trainer VALUES (1, 'Derek', 'derek@fitness.me', 'Vancouver', 'Canada');
        INSERT INTO Trainer VALUES (2, 'Shawn', 'shawn@fitness.me', 'Berlin', 'Germany');
        INSERT INTO Trainer VALUES (3, 'Greg', 'greg@fitness.me', 'New York', 'USA');
        INSERT INTO Trainer VALUES (4, 'Chris', 'chris@fitness.me', 'Tampa', 'USA');
        INSERT INTO Trainer VALUES (5, 'Jeff', 'jeff@fitness.me', 'Kelowna', 'Canada');
```

Personal Reminders

```
        INSERT INTO PersonalReminders VALUES (1, 1, 'Biweekly', 'Take a break');
        INSERT INTO PersonalReminders VALUES (2, 2, 'Weekly', 'Let us work out');
        INSERT INTO PersonalReminders VALUES (3, 3, 'Daily', 'Go for a run');
        INSERT INTO PersonalReminders VALUES (4, 4, 'Monthly', 'Start exercising');
        INSERT INTO PersonalReminders VALUES (5, 5, 'Weekly', 'Take a break');
```

NutritionPlan1

```
INSERT INTO NutritionPlan1 VALUES (1, 100, 100, 100);
INSERT INTO NutritionPlan1 VALUES (2, 100, 100, 100);
INSERT INTO NutritionPlan1 VALUES (3, 300, 300, 300);
INSERT INTO NutritionPlan1 VALUES (4, 100, 200, 300);
INSERT INTO NutritionPlan1 VALUES (5, 0, 0, 0);
```

NutritionPlan2

```
INSERT INTO NutritionPlan2 VALUES (100, 100, 100, 300);
INSERT INTO NutritionPlan2 VALUES (200, 200, 200, 600);
INSERT INTO NutritionPlan2 VALUES (100, 200, 300, 600);
INSERT INTO NutritionPlan2 VALUES (0, 0, 0, 0);
INSERT INTO NutritionPlan2 VALUES (300, 300, 300, 900);
```

ExercisePlan

```
        INSERT INTO ExercisePlan VALUES(1, 'Beginner', 1, 2);
        INSERT INTO ExercisePlan VALUES(2, 'Beginner 2.0', 1, 3);
        INSERT INTO ExercisePlan VALUES(3, 'Mr. Olympia', 2, 2);
        INSERT INTO ExercisePlan VALUES(4, 'Intermediate', 3, 4);
        INSERT INTO ExercisePlan VALUES(5, 'Marathon', 1, 5);
```

Exercise

```
INSERT INTO Exercise VALUES (1, 'Bodybuilding', 'Bench Press');
INSERT INTO Exercise VALUES (2, 'Bodybuilding', 'Front Squat');
INSERT INTO Exercise VALUES (3, 'Cardio', 'Swimming');
INSERT INTO Exercise VALUES (4, 'Endurance', 'HIIT');
INSERT INTO Exercise VALUES (5, 'Mindfulness', 'Yoga');
```

Content

```
INSERT INTO Content VALUES (1, 'Mike Mentzer', 'www.fitnessapp.com/article1');
INSERT INTO Content VALUES (2, 'Arnold Schwarzenegger', 'www.fitnessapp.com/article2');
INSERT INTO Content VALUES (3, 'Andrew Huberman', 'www.fitnessapp.com/article3');
INSERT INTO Content VALUES (7, 'LeFraud James', 'www.fitnessapp.com/article4');
INSERT INTO Content VALUES (4, 'Jeff Nippard', 'www.fitnessapp.com/article6');
```

Includes

```
INSERT INTO Includes VALUES (1,1);
INSERT INTO Includes VALUES (1,2);
INSERT INTO Includes VALUES (1,3);
INSERT INTO Includes VALUES (2,1);
INSERT INTO Includes VALUES (2,2);
```