University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 1

Date: Mar 1, 2023

Group Number: 25

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Bryan Hui	53984746	x8f3l	bryanhui77@hotmail.com
Dhrubo Karmaker	13198379	w5u9r	dhruv.karmaker@gmail.com
Harper Kim	45579521	p0f7q	wowhkk@gmail.com

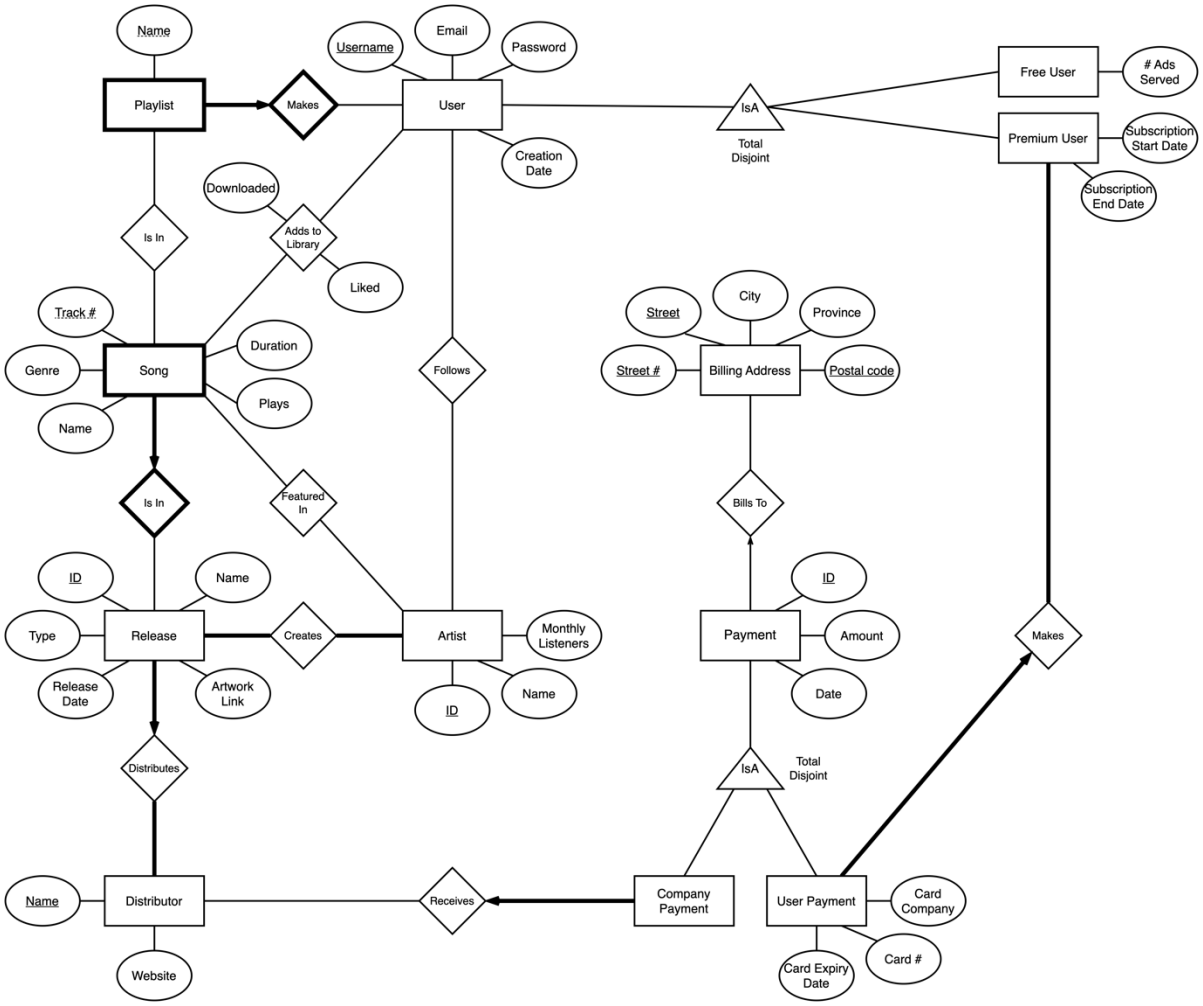
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 1, the main purpose of this page is for you to let us know your e-mail address, 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.

Project Summary

Our project reflects and models streaming platforms in the music industry such as Spotify, Apple Music, or YouTube Music. Our database will allow companies to manage content interactions and payments between users, releases, artists, and distributors in their streaming services.
(Note: Our database only manages payments and transactions that occur in Canada.)

ER Diagram & Changes



We have mostly overhauled the payment and subscription-related aspects of our project.

- The Subscription Plan entity, which represented multiple tiers, was replaced with an IsA of two user types: Free User and Premium User. The database now keeps track of the number of ads a Free User receives. We believe that this change models a streaming service more closely and allows better management of subscription tiers.
- We have removed the splitting of payments between Distributor and Artist, as we believe this is the responsibility of the distribution company (like in real life). All payments now go to Distributor directly.

- Payment is now a general entity that represents any type of transaction that occurs. There are two types of Payments, represented by an IsA. The User Payment records Premium User transactions and stores credit card information. The Company Payment records information about payments to Distributors. Additionally, all payments now have a new attribute, Billing Address.
- We have also made some minor adjustments throughout the diagram. The User entity now has several new attributes: Email, Password, and Creation Date. The Genre entity was changed into an attribute of Song. We removed the IsA for Distributor as it was not very significant.
- Purposely created redundancies to allow for non-primary key functional dependencies (see below)

Schema

**Bolded* means foreign key*
**Underlined* means primary key*
**CK means candidate key*

**Note: general attribute names are written with its entity's name. E.g. Name in distributor is DistributorName, ID in Release is ReleaseID*

Release(ID: int, Name: char(50), Type: char(6), ReleaseDate: date, ArtURL: char(150), **DistributorName: char(150) (not null)**)

Song(**ReleaseID: int**, TrackNum: int, Name: char(50), Duration: int, Genre: char(50), Plays: int)

Artist(ID: int, Name: char(50), MonthlyListeners: int)

Distributor(Name: char(150), Website: char(50))

UserPayment(ID: int, Amount: numeric(10, 2), Date: date, CardCompany: char(50), CardNum: int, CardExpiry: date, **Username: char(50) (not null), StreetNum: int, Street: char(50), PostalCode: char(6)**)

CompanyPayment(ID: int, Amount: numeric(10, 2), Date: date, **DistributorName: char(150) (not null), StreetNum: int, Street: char(50), PostalCode: char(6)**)

User(Username: char(50), Email: char(50) (CK), Password: char(50), CreationDate: date)
(Candidate keys: Username, Email)

FreeUser(**Username: char(50)**, AdsServed: integer)

PremiumUser(**Username: char(50)**, SubStartDate: date, SubEndDate: date)

Playlist(**Username: char(50)**, Name: char(50))

BillingAddress(StreetNum: int, Street: char(50), City: char(50), Province: char(2), PostalCode: char(6))

PlaylistIsIn(**Username: char(50), Name: char(50), TrackNum: int, ReleaseID: int**)

AddsToLibrary(**ReleaseID: int**, **TrackNum: int**, **Username: char(50)**, Downloaded: numeric(1,0), Liked: numeric(1,0))

FeaturedIn(**ArtistID: int, ReleaseID: int**, **TrackNum: int**)

Follows (**Username: char(50), ArtistID: int)**

Creates (**ArtistID: int, ReleaseID: int)**

Functional Dependencies

Song Table:
ReleaseID, TrackNum → Genre, Duration, Name, Plays

Release Table:
ID → Type, ReleaseDate, ArtURL, Name, DistributorName

Artist Table:
ID → Name, MonthlyListeners

Distributor Table:
Name → Website

UserTable:
Username → Email, Password, CreationDate
Email → Username, Password, CreationDate

FreeUser Table:
Username → AdsServed
PremiumUser Table:
Username → SubStartDate, SubEndDate

UserPayment Table:
ID → Amount, Date, Card Company, CardNum, CardCompany, CardExpiry, PostalCode, Street, StreetNum, Username
Non PK: CardNum → CardCompany, CardExpiry

CompanyPayment Table:
ID → Amount, Date, StreetNum, Street, PostalCode, DistributorName

BillingAddress Table:
StreetNum, Street, PostalCode → City, Province
Non PK: PostalCode → City, Province

AddsToLibrary Table:
ReleaseID, TrackNum, Username → Downloaded, Liked

Normalization

BillingAddress Table:

BillingAddress(StreetNum: int, Street: char(50), City: char(50), Province: char(2), PostalCode: char(6))

PostalCode → City,Province

PostalCode+ = {City,Province,PostalCode}

Therefore PostalCode is not a superkey and the table needs to be decomposed.

Decomposed form:

BillingAddress(StreetNum: int, Street: char(50), **PostalCode: char(6)**)

PostalCodeCityProvince(City: char(50), Province: char(2), PostalCode: char(6))

UserPayment Table:

UserPayment(ID: int, Amount: numeric(10, 2), Date: date, CardCompany: char(50), CardNum: int, CardExpiry: date, **Username: char(50) (not null), StreetNum: int, Street: char(50), PostalCode: char(6)**)

CardNum → CardCompany, CardExpiry

CardNum+ = {CardNum, CardCompany, CardExpiry}

So, CardNum is not a superkey. UserPayment table needs to be decomposed.

Decomposed form:

UserPayment(ID: int, Amount: numeric(10, 2), Date: date, **CardNum: int**, **Username: char(50) (not null), StreetNum: int, Street: char(50), PostalCode: char(6)**)

CardTable(CardCompany: char(50), CardNum: int, CardExpiry: date)

Other tables are normalized and in 3NF.

SQL DDL

(Note: Our project will likely use Oracle, so `ON UPDATE`s are excluded.)

```
CREATE TABLE Release
    (ID                  INT PRIMARY KEY,
Name                VARCHAR(50),
Type                VARCHAR(6),
ReleaseDate         DATE,
ArtURL              VARCHAR(150),
DistributorName     VARCHAR(150) NOT NULL,
FOREIGN KEY (DistributorName)
          REFERENCES Distributor(Name)
          ON DELETE CASCADE);

CREATE TABLE Song
    (ReleaseID INT,
TrackNum  INT,
Name      VARCHAR(50),
Duration  INT,
Genre     CHAR(50),
Plays     INT,
PRIMARY KEY (ReleaseID, TrackNum),
FOREIGN KEY (ReleaseID)
          REFERENCES Release(ID)
          ON DELETE CASCADE);

CREATE TABLE Artist
    (ID                  INT PRIMARY KEY,
Name                VARCHAR(50),
MonthlyListeners    INT);

CREATE TABLE Distributor
    (Name      VARCHAR(150) PRIMARY KEY,
Website   VARCHAR(50));

CREATE TABLE UserPayment
    (ID             INT,
     Amount         NUMERIC(10, 2),
     Date           DATE,
     CardNum        INT,
     Username       VARCHAR(50) NOT NULL,
```

```
     StreetNum      SMALLINT,
     Street         CHAR(50),
     PostalCode     CHAR(6),
     PRIMARY KEY(ID),
     FOREIGN KEY (CardNum)
          REFERENCES CardTable(CardNum)
          ON DELETE CASCADE,
     FOREIGN KEY (Username)
          REFERENCES PremiumUser(Username)
          ON DELETE CASCADE,
     FOREIGN KEY (StreetNum, Street, PostalCode)
          REFERENCES BillingAddress
               (StreetNum, Street, PostalCode)
          ON DELETE CASCADE);
CREATE TABLE CardTable
    (CardCompany    CHAR(50),
CardNum        INT PRIMARY KEY,
CardExpiry     DATE);

CREATE TABLE CompanyPayment
    (ID             INT PRIMARY KEY,
     Amount         NUMERIC(10, 2),
     Date           DATE,
     StreetNum      SMALLINT,
     Street         CHAR(50),
     PostalCode     CHAR(6),
     DistributorName     VARCHAR(150) NOT NULL,
     FOREIGN KEY (DistributorName)
          REFERENCES Distributor(Name)
          ON DELETE CASCADE,
     FOREIGN KEY (StreetNum,Street,PostalCode)
        REFERENCES BillingAddress(StreetNum,Street,PostalCode)
          ON DELETE CASCADE);

CREATE TABLE BillingAddress
    (StreetNum      SMALLINT,
     Street         VARCHAR(50),
     PostalCode     VARCHAR(6),
     PRIMARY KEY (StreetNum, Street, PostalCode)
     FOREIGN KEY (PostalCode)
          REFERENCES PostalCodeCityProvince(PostalCode)
          ON DELETE CASCADE);

CREATE TABLE PostalCodeCityProvince
    (City           VARCHAR(50),
```

```
Province       VARCHAR(2),
PostalCode     VARCHAR(6) PRIMARY KEY);

CREATE TABLE User
    (Username       VARCHAR(50) PRIMARY KEY,
     Email          VARCHAR(50) UNIQUE,
     Password       VARCHAR(50),
     CreationDate   DATE);

CREATE TABLE FreeUser
    (Username       VARCHAR(50) PRIMARY KEY,
AdsServed      INT,
     FOREIGN KEY (Username)
          REFERENCES User(Username)
          ON DELETE CASCADE);


CREATE TABLE PremiumUser
    (Username       VARCHAR(50) PRIMARY KEY,
     SubStartDate   DATE,
     SubEndDate     DATE,
     FOREIGN KEY (Username)
          REFERENCES User(Username)
          ON DELETE CASCADE);

CREATE TABLE Playlist
    (Username  VARCHAR(50),
Name      VARCHAR(50),
PRIMARY KEY (Username, Name),
FOREIGN KEY (Username)
          REFERENCES User(Username)
          ON DELETE CASCADE);

CREATE TABLE PlaylistIsIn
    (Username  VARCHAR(50),
Name      VARCHAR(50),
ReleaseID INT,
TrackNum  INT,
PRIMARY KEY (Username, Name, TrackNum, ReleaseID),
FOREIGN KEY (Username, Name)
          REFERENCES Playlist(Username, Name)
          ON DELETE CASCADE,
     FOREIGN KEY (TrackNum, ReleaseID)
          REFERENCES Song(TrackNum, ReleaseID)
          ON DELETE CASCADE);
```

```
CREATE TABLE AddsToLibrary
    (ReleaseID INT,
TrackNum  INT,
Username  VARCHAR(50),
DOWNLOADED NUMERIC(1,0),
LIKED NUMERIC(1,0),
PRIMARY KEY (ReleaseID, TrackNum, Username),
FOREIGN KEY (Username)
          REFERENCES Song (ReleaseID, TrackNum)
          ON DELETE CASCADE,
FOREIGN KEY (Username)
          REFERENCES User(Name)
          ON DELETE CASCADE);

CREATE TABLE FeaturedIn
    (ArtistID  INT,
ReleaseID INT,
TrackNum  INT,
PRIMARY KEY (ArtistID, ReleaseID, TrackNum),
FOREIGN KEY (ArtistID)
          REFERENCES Artist(ID)
          ON DELETE CASCADE,
FOREIGN KEY (ReleaseID, TrackNum)
          REFERENCES Song(ReleaseID, TrackNum)
          ON DELETE CASCADE);

CREATE TABLE Follows
    (Username  VARCHAR(50),
ArtistID  INT,
PRIMARY KEY (Username, ArtistID),
FOREIGN KEY (Username)
          REFERENCES User(Username)
          ON DELETE CASCADE,
FOREIGN KEY (ArtistID)
          REFERENCES Artist(ID)
          ON DELETE CASCADE);

CREATE TABLE Creates
    (ReleaseID INT,
ArtistID  INT,
PRIMARY KEY (ReleaseID, ArtistID),
FOREIGN KEY (Username)
          REFERENCES Artist (ID)
          ON DELETE CASCADE,
```

```
FOREIGN KEY (ReleaseID)
          REFERENCES Release(ID)
          ON DELETE CASCADE);
```

SQL INSERT

```
INSERT INTO Release VALUES
(1, 'Whenever You Need Somebody', 'Album', 1987-11-12,
'https://cdn.spotube.com/img/rickroll.png', 'BMG Rights
Management (UK) Limited'),
(2, 'Astro Lounge', 'Album', 1999-06-08,
'https://cdn.spotube.com/img/shrek.png', 'Interscope Geffen
(A&M) Records'),
(3, 'Curb Your Enthusiasm (Music From the TV Series)', 'Album',
2006-04-17, 'https://cdn.spotube.com/img/frolic.png',
'Mellowdrama Records'),
(4, 'Tritio Jatra', 'Album', 2006-04-17,
'https://cdn.spotube.com/img/nemesis.png', 'Studio Bangi'),
(5, 'Superunknown', 'Album', 2006-04-17,
'https://cdn.spotube.com/img/soundgarden.png', 'Sony Music');

INSERT INTO Song VALUES
(1, 1, 'Never Gonna Give You Up', 214, 'Pop', 213422997),
(1, 2, 'Whenever You Need Somebody', 243, 'Pop', 1842825),
(2, 4, 'All Star', 190, 'Alternative', 692894289),
(3, 1, 'Frolic', 210, 'Soundtrack', 5359039),
(4, 4, 'Nirbashon', 210, 'Rock', 5359039);

INSERT INTO Artist VALUES
(1, 'Rick Astley', 5829256),
(2, 'Smash Mouth', 3950387),
(3, 'Curb Your Enthusiasm', 38358),
(4, 'Luciano Michelini', 35294);
(5, 'Nemesis', 200000);

INSERT INTO Distributor VALUES
('BMG Rights Management (UK) Limited', 'https://www.bmg.com/'),
('Interscope Geffen (A&M) Records',
'https://www.interscope.com'),
('Mellowdrama Records', NULL);
('Studio Bangi', ' https://www.studio-bangi.com');
('Sony Music', 'https://www.sonymusic.com');

INSERT INTO UserPayment VALUES
(1, 7, 2022-03-01, 2394235323332438, 'bryanh', 3235, 'Joyce
st.', 'V6E7F3'),
```

```
(2, 11, 2022-03-01, 5434235323332438, 'sleepy', 1234, 'Joyce
st.', 'V6E7F3'),
(3, 12, 2022-02-01, 1234353233324332, 'A113', 1234, 'Cambie
st.', 'V6E7F3'),
(4, 15, 2022-02-01, 7893353233324342, 'notstevejobs', 1234,
'Cambie st.', 'V4E7F3'),
(5, 15, 2022-02-01, 8989353233324023, 'celloist', 7843, 'Cambie
st.', 'V6E7F3');

INSERT INTO CardTable VALUES
('Visa', 2394235323332438, 2025-01-01),
('Amex', 5434235323332438, 2026-01-01),
('Mastercard', 1234353233324332, 2027-01-01),
('Visa', 7893353233324342, 2025-02-01),
('Amex', 8989353233324023, 2023-07-01);

INSERT INTO CompanyPayment VALUES
(1, 7000, 2022-03-01, 3235, 'Courtney St.', 'V6E5E3', 'Sony
Music'),
(2, 1100, 2022-03-01, 1234, 'Camosun St.', 'V6E789', 'Studio
Bangi'),
(3, 1200, 2022-02-01, 1234, 'Cambie st.', 'V8E7F3', 'BMG Rights
Management (UK) Limited'),
(4, 1500, 2022-02-01, 1234, 'Hastings st.', 'V6E7K3',
'Interscope Geffen (A&M) Records'),
(5, 1500, 2022-02-01, 7843, 'Manitoba st.', 'V6E1K2',
'Mellowdrama Records');

INSERT INTO BillingAddress VALUES
(3235, 'Joyce st.', 'V6E7F3'),
(1234, 'Joyce st.', 'V6E7F3'),
(1234, 'Cambie st.', 'V4E7F3'),
(2829, 'Cambie st.', 'V6E7F3'),
(7843, 'Cambie st.', 'V6E7F3'),
(3235, 'Courtney St.', 'V6E5E3'),
(1234, 'Camosun St.', 'V6E789'),
(2948, 'Cambie st.', 'V4E7F3'),
(1234, 'Hastings st.', 'V8E7K3',),
(7843, 'Manitoba st.', 'V6E1K2');

INSERT INTO PostalCodeCityProvince VALUES
('Vancouver', 'BC', 'V6E7F3'),
('Vancouver', 'BC', 'V4E7F3'),
('Vancouver', 'BC,' 'V6E5E3'),
('Vancouver', 'BC', 'V6E789'),
('Vancouver', 'BC', 'V8E7K3'),
('Vancouver', 'BC', 'V6E7K3'),
```

```
('Vancouver', 'BC', 'V6E1K2');

INSERT INTO User VALUES
('bryanh', 'bryan@gmail.com', 'CS213Sucks', 2022-01-01),
('harperk', 'harper@gmail.com', 'CS310Sucks', 2022-02-02),
('dhrubok', 'dhrubo@hotmail.com', 'CS210Sucks', 2021-09-01),
('kimdol', 'kimdol@yahoo.com', 'CS304Sucks', 2022-03-04),
('A113, 'A113@outlook.com', 'CS304Sucks', 2022-09-10),
('eenie meenie', 'eenie@gmail.com', 'whatever', 2021-02-09),
('sleepy', 'sleepingInClass@gmail.com', 'allnighter', 2023-01-
01),
('celloist', 'cello-ing@yahoo.com', 'prez', 2021-04-06),
('notstevejobs', 'jobs@gmail.com', 'googlebest', 2020-01-09),
('jfk1975', 'jfk@yahoo.com', 'ishouldbedead', 2019-01-03);

INSERT INTO FreeUser VALUES
('harperk', 13),
('dhrubok', 16),
('kimdol', 0),
('jfk1975', 13),
('eenie meenie', 6);

INSERT INTO PremiumUser VALUES
('bryanh', 2022-03-01, 2023-03-01),
('A113', 2022-02-01, 2023-02-01),
('sleepy', 2023-03-01, 2024-02-01),
('celloist', 2023-03-01, 2024-02-01),
('notstevejobs', 2021-03-01, 2022-02-01);

INSERT INTO Playlist VALUES
('bryanh','lofi'),
('eenie meenie','90s jam',),
('kimdol','study-session'),
('jfk1975','80s glam metal'),
('A113','alt jam');

INSERT INTO PlaylistIsIn VALUES
('bryanh','lofi',1,1),
('bryanh','lofi',1,2),
('kimdol','study-session',2,4),
('eenie meenie',' 90s jam',4,4),
('jfk1975','80s glam metal',3,1);

INSERT INTO AddsToLibrary VALUES
(1,1,'bryanh',0,1),
(1,2,'bryanh',0,0),
(2,4,'kimdol',0,1),
```

```
(4,4,'sleepy',1,0),
(3,1,'A113',1,1);

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

INSERT INTO Follows VALUES,
('bryanh',1),
('kimdol',2),
('eenie meenie',3),
('A113','notstevejobs',2),
('sleepy',5);

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