

CPSC 304 Project Cover Page

Milestone #: 2

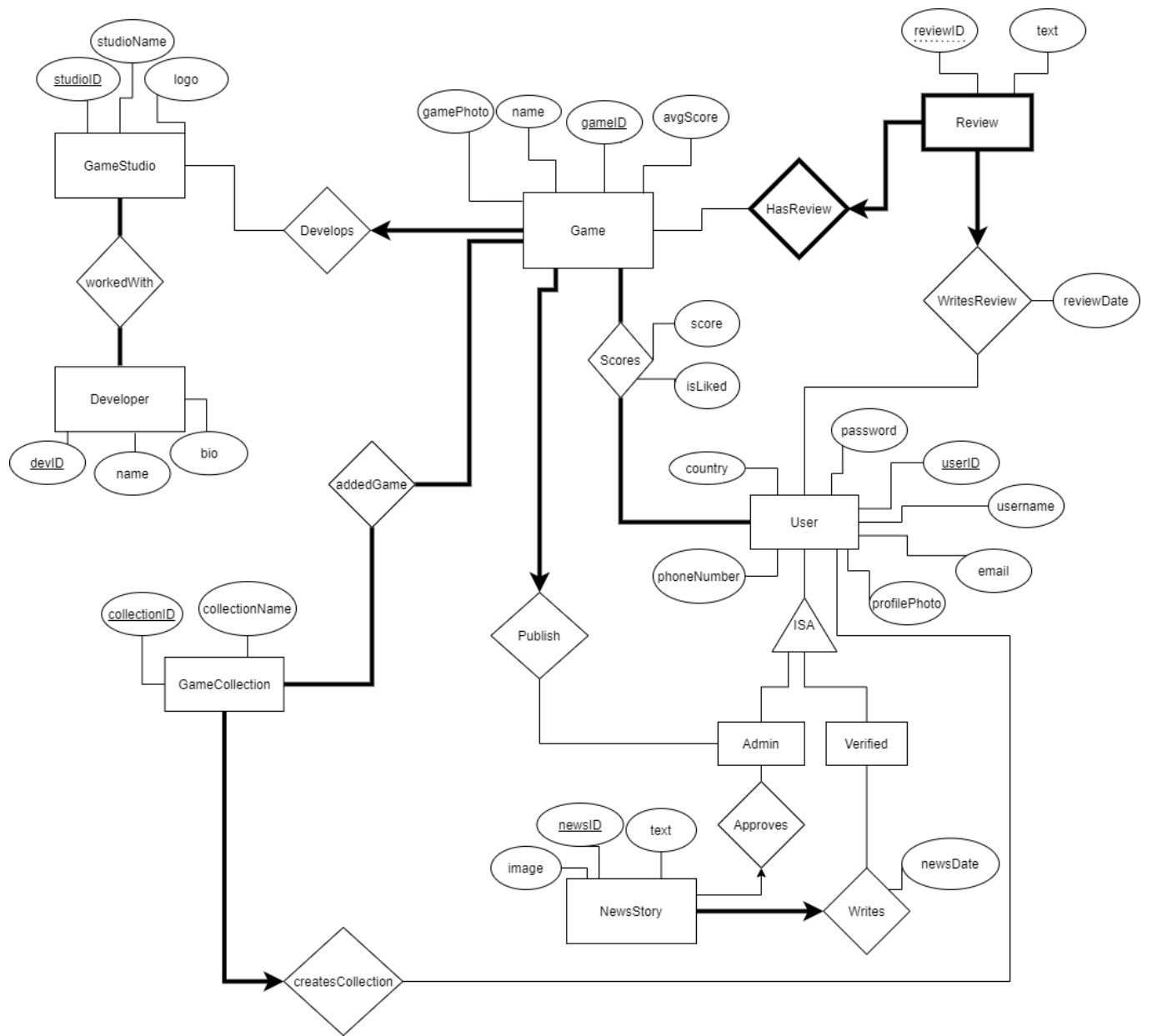
Date: 10.25.2021

Group Number: 15

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Kiril Lazarev	93286631	y9a3b	klazarev2000gmail.com
Baris Ozkan	85984987	l3w2b	bozkanubc@gmail.com
Efe Demir	88882931	u1w2b	efedemir2001@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



#3 Schema:

General constraint: all IDs are 8 characters long

GameStudio (studioID: integer, studioName: varchar, logo: image)

Developer (devID: integer, devName: varchar, bio: varchar)

workedWith (studioID: integer, devID: integer)

Game (gameID: integer, **adminID**: integer, **studioID**: integer, gameName: varchar, gamePhoto: image, avgScore: float)

Constraint: avgScore is a float between 0 - 10

User (userID: integer, username: varchar, profilePhoto: image, email: varchar, phoneNumber: varchar, country: varchar, password: varchar)

Candidate Keys: userID, email

Verified(userID: integer)

Admin(userID: integer)

GameCollection(collectionID: integer, **userID**: integer, collectionName: varchar)

Review (reviewID: integer, **gameID**: integer, **userID**: integer, text: varchar, reviewDate: date)

Scores (**gameID**: integer, userID: integer, score: float, isLiked: boolean)

Constraint: score attribute is a float between 0 - 10

NewsStory (newsID: integer, **verifiedUserID**: integer, **adminUserID**: integer, text: varchar, image: image, newsDate: date)

addedGame (collectionID: integer, **gameID**: integer)

#4 Functional Dependencies

Assumption: logo and gamePhoto determine name because of copyright laws, but the same profile photos can be used by many users. For example, the Riot Games logo implies the GameStudio's name.

GameStudio:	studioID -> (studioName, logo) studioName-> (logo)
Developer:	devID -> (name, bio)
Game:	gameID -> (gameName, studioID, adminUserID, gamePhoto, avgScore) gameName-> (gamePhoto)
Review:	reviewID -> (text, gameID, reviewDate, userID)
NewsStory:	newsID -> (text, image, verifiedUserID, adminUserID, newsDate)
User:	userID -> (username, profilePhoto, collectionID, reviewID, country, phoneNumber) phoneNumber -> (country)
Scores:	gameID, userID -> (score, isLiked)
GameCollection:	collectionID -> (collectionName, userID)

Relations being decomposed:

```
{studioID}+ = studioID, studioName, logo
{studioName}+ = logo, studioName
```

```
GameStudio(studioID, studioName)
studioLogo(studioName, logo)
```

```
{gameID}+ = gameID, name, gamePhoto, avgScore, adminUserID, studioID
{name}+ = gamePhoto, name
```

```
Game(gameID, name, avgScore, adminUserID, studioID)
name(name, gamePhoto)
```

```
{userID}+ = userID, username, profilePhoto, collectionID, reviewID, country,
                                     phoneNumber
{phoneNumber}+ = phoneNumber, country
```

```
User(username, profilePhoto, collectionID, reviewID, phoneNumber)
UserPhone(phoneNumber, country)
```

Normalized Schema

General constraint: all IDs are 8 characters long

GameStudio (studioID: integer, name: varchar)

StudioLogo(studioName: varchar, logo: image)

Developer (devID: integer, devName: varchar, bio: varchar)

workedWith (studioID: integer, devID: varchar)

User (userID: integer, username: varchar, profilePhoto: image, email: varchar, password: varchar, phoneNumber: varchar)

Candidate Keys: userID, email

UserPhone(phoneNumber: varchar, country: varchar)

Verified(userID: integer)

Admin(userID: integer)

GameCollection(collectionID: integer, collectionName: varchar, userID: varchar)

Review (reviewID: integer, gameID: integer, userID: integer, text: varchar, reviewDate: date)

Scores (gameID: integer, userID: integer, score: float, isLiked: boolean)

Constraint: score attribute is a float between 0 - 10

Game (gameID: integer, adminUserID: integer, studioID: integer, avgScore: float, gameName: varchar)

PublishedPhoto(gameName: varchar, gamePhoto: image)

NewsStory (newsID: integer, verifiedUserID: integer, adminUserID: integer, text: varchar, image: image, newsDate: date)

addedGame (collectionID: integer, gameID: integer)

#6 SQL DDL

```
CREATE TABLE GameStudio
```

```
(
  studioID INTEGER,
  studioName VAR CHAR NOT NULL UNIQUE,
  PRIMARY KEY (studioID)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```

```
CREATE TABLE StudioLogo
```

```
(
  studioName VARCHAR,
  brandLogo Image NOT NULL,
  PRIMARY KEY (studioName),
  FOREIGN KEY (studioName) REFERENCES GameStudio
);
```

```
CREATE TABLE Developer
```

```
(
  devID INTEGER,
  name VARCHAR NOT NULL,
  bio VARCHAR NOT NULL,
  PRIMARY KEY (devID)
);
```

```
CREATE TABLE workedWith
```

```
(
  devID INTEGER,
  studioID INTEGER,
  PRIMARY KEY (devID, studioID)
  FOREIGN KEY (devID) REFERENCES Developer
  ON DELETE CASCADE,
  FOREIGN KEY (studioID) REFERENCES Studio
  ON DELETE CASCADE,
);
```

```
CREATE TABLE Game
```

```
(
  gameID INTEGER,
  studioID INTEGER NOT NULL,
  adminUserID INTEGER NOT NULL,
  gameName VARCHAR NOT NULL UNIQUE
  ON UPDATE CASCADE,
  avgScore FLOAT NOT NULL,
  PRIMARY KEY (gameID),
  FOREIGN KEY (studioID) REFERENCES GameStudio,
  FOREIGN KEY (adminUserID ) REFERENCES Admin
);
```

```
CREATE TABLE PublishedPhoto
(
  gameName VARCHAR,
  gamePhoto Image NOT NULL,
  PRIMARY KEY (gameName) REFERENCES Game,
  FOREIGN KEY (gameName) REFERENCES Game
);
```

```
CREATE TABLE Review
(
  reviewID INTEGER,
  text VARCHAR(300) NOT NULL,
  userID INTEGER NOT NULL,
  gameId INTEGER NOT NULL,
  reviewDate DATE,
  PRIMARY KEY (reviewID),
  FOREIGN KEY (userID) REFERENCES (User)
  ON DELETE SET NULL,
  FOREIGN KEY (gameID) REFERENCES (Game)
  ON DELETE SET NULL,
);
```

```
CREATE TABLE User
(
  userID INTEGER,
  username VARCHAR(20) NOT NULL
  ON UPDATE CASCADE,
  profilePhoto Image NOT NULL,
  email VARCHAR(20) NOT NULL,
  phoneNumber varchar,
  PRIMARY KEY (userID)
);
```

```
CREATE TABLE UserPhone
(
  phoneNumber varchar,
  country varchar,
  PRIMARY KEY (phoneNumber),
  FOREIGN KEY (phoneNumber) REFERENCES User (phoneNumber)
);
```

```
CREATE TABLE Scores
(
  userID INTEGER NOT NULL,
  gameId INTEGER NOT NULL,
  score FLOAT,
  isLiked BOOLEAN,
  PRIMARY KEY (userID, gameId),
  FOREIGN KEY (userID) REFERENCES User,,
  FOREIGN KEY (gameID) REFERENCES Game
);
```



```
CREATE TABLE Verified
(
  userID VARCHAR(8) NOT NULL,
  PRIMARY KEY (userID),
  FOREIGN KEY (userID) REFERENCES User(userID),
);
```

```
CREATE TABLE Admin
(
  userID VARCHAR(8) NOT NULL,
  PRIMARY KEY (userID),
  FOREIGN KEY (userID) REFERENCES User(userID)
);
```

```
CREATE TABLE NewsStory
(
  newsID VARCHAR(8) NOT NULL,
  verifiedUserID INTEGER NOT NULL,
  adminUserID INTEGER,
  text VARCHAR(300) NOT NULL,
  image Image NOT NULL,
  newsDate DATE,
  PRIMARY KEY (newsID),
  FOREIGN KEY (verifiedUserID) REFERENCES (Verified),
  FOREIGN KEY (adminUserID) REFERENCES (Admin)
);
```

```
CREATE TABLE gameCollection
(
  collectionID INTEGER,
  userID INTEGER NOT NULL,
  name VARCHAR(30) NOT NULL,
  PRIMARY KEY (collectionID),
  FOREIGN KEY (userID) REFERENCES User
);
```

```
CREATE TABLE addedGame
(
  gameID INTEGER NOT NULL,
  collectionID VARCHAR(8) NOT NULL,
  PRIMARY KEY (gameID , collectionID),
  FOREIGN KEY (gameID ) REFERENCES Game,
  FOREIGN KEY (collectionID) REFERENCES GameCollection(collectionID)
);
```

Part 7										
	GameStudio			StudioLogo			Developer			
	<u>studioID</u>	name		<u>name</u>	image		<u>devID</u>	name	bio	
	90000001	Steam		Steam	steam.jpeg		80000001	Gaben Newell	mmm Money	
	90000002	Electronic Arts		Electronic Arts	ea.png		80000002	Joe Biden	America!	
	90000003	Riot Games		Riot Games	riotgames.jpeg		80000003	Kim Kardashian	omg i love games	
	90000004	Blizzard Entertainment		Blizzard Entertainment	blizzard.png		80000004	Nigel Thornberry	smashing!!	
	90000005	Epic Games		Epic Games	epic.jpeg		80000005	Stich	hi	
	workedWith			User						
	<u>studioID</u>	<u>devID</u>		<u>userID</u>	username	profilePhoto	email	password	phoneNumber	
	90000001	80000005		71000001	kuluku	riley.jpeg	k@gmail.com	123456789	(+1) 123456789	
	90000002	80000003		71000002	Wolfenshroud	anime.png	w@hotmail.com	123456789	(+90) 23245453	
	90000003	80000001		71000003	Kid&Eel	orangutan.png	ke@ubc.ca	iamaverycoolguy	(+31) 34234234	
	90000004	80000004		70000004	n.i.m.b	nimb.jpeg	n@google.com	twoplustwoequals4	(+2) 234234324	
	90000005	80000002		70000005	the_legend_27	dog.png	27@gmail.com	!@!@hi_hi@!!@	(+99) 111111111	
	UserPhone			Verified			Admin		GameCollection	
	<u>phoneNumber</u>	country		<u>userID</u>		<u>userID</u>		<u>collectionID</u>	collectionName	<u>userID</u>
	(+1) 123456789	Canada		70000001		71000001		600000001	myFavoriteGames	71000001
	(+90) 23245453	Turkey		70000002		71000002		600000002	bestFPS	70000002
	(+31) 34234234	Netherlands		70000003		71000003		600000003	top10games	70000005
	(+2) 234234324	Faroe Islands		70000004		71000004		600000004	i_hate_these_games	70000004
	(+99) 111111111	Iraq		70000005		71000005		600000005	jotaro	71000002
	Review						Scores			
	<u>reviewID</u>	<u>gameID</u>	<u>userID</u>	text	reviewDate		<u>gameID</u>	<u>userID</u>	score	isLiked
	50000001	40000001	71000001	very nice game	10/10/2021		40000001	71000001	0	FALSE
	50000002	40000002	71000002	bad game	1/1/2020		40000002	71000002	2	FALSE
	50000003	40000003	71000003	ok game.	2/2/2022		40000003	71000003	8	TRUE
	50000004	40000004	70000004	10/10	3/3/2023		40000004	70000004	8	FALSE
	50000005	40000005	70000005	0/10 garbage!	5/5/2025		40000005	70000005	10	TRUE
	Game						gamePhoto			
	<u>gameID</u>	<u>adminUserID</u>	<u>studioID</u>	name	avgScore		<u>name</u>	gamePhoto		
	40000001	71000001	90000001	CS:GO	8.8		CS:GO	csgo.jpeg		
	40000002	71000002	90000002	FIFA	4.0		FIFA	fifa.png		
	40000003	71000003	90000003	League of Legends	7.6		League of Legends	league.jpeg		
	40000004	71000004	90000004	Overwatch	3.1		Overwatch	overwatch.png		
	40000005	71000005	90000005	Fortnite	6.9		Fortnite	fortnite.jpeg		

#8 Queries In Plain English:

1. Insertion: Add a user id, username, profile photo, email, password and phoneNumber to User
2. Deletion: remove user id and all its corresponding attributes from User, which sets Review entity attached to user id to null.
3. Update: update username, cascades through possible relationships with newstories or published games
4. Insertion: Add devID, name, and bio to Developer
5. Deletion: Delete developer from Developer, the relationship between that developer and studio is also deleted
6. Insertion: Add studioID, studioName, and logo to GameStudio
7. Deletion: Delete game studio from GameStudio set, the relationship workedWith between developers and that studio is deleted.
8. Insertion: Add gameID, studioID, gamePhoto, and name to Game
9. Update: Updating game's name causes all games with that game ID to change
10. Deletion: Delete gameID and its corresponding attributes from Game, and set Review's game ID to null
11. Insertion: Add newsID, image, text, adminID, and verifiedUserID to NewsStory
12. Deletion: Deletes newsID and its corresponding attributes from NewsStory