PHASE 3 – IMPLEMENTATION

By: Lucas Gullo, Geoffrey Splendorio, Randi Tinney

COMP 4410 Database Management

Mahmood Hossain

November 29, 2018

**Purpose:**

We are to create a program for the company, *Movies-R-Us*. We are tasked to design a backend database for the company’s users and inventory. The users will be able to login to the front end Java application with their registered email address and their password. Once in they will be able to see all of the carried games and movies and, depending on the member level they paid for, will be able to rent a certain number of the available forms of entertainment. They will be able to search movies by actor, director, genre, sequels, and keywords. They will also be able to filter by award winning movies and movies that they haven’t checked out previously. They will be able to search games by genre, platform, and keywords. Finally, they will be able to see their rent history which will include all of the details of the movie/game, check out dates, and return dates.

There will also be specific users called administrators that will be able to add/remove members, movies, and games. They will also be able to update inventory details. They will be able to view the items that were rented in the last 24hours which will include the order details and shipping information. Finally, they will be able to see the top ten items based on number of rentals for the past month.

Within the database, the inventory will include movies and games. Each movie within the database will include the following details: title, cast, director, release date, genre, awards won, and the immediate sequel if one exists. Information for all of the cast and director will include their name and address. Each game within the database will include the following details: title, release date, genre, platform, and version. Finally, each game and movie will include the total number of copies available and the current number of copies available.

Within the database, each user will include the following information: email, name, address, phone#, password, their quota for rentals, and past rental history.

For this phase, we were tasked with creating a conceptual design of the back end database via an ER diagram which details the various entities, attributes, relations, and the constraints that we feel are necessary for this project.

**Assumptions:**

For this project, we assumed that an administrator would automatically be a member by design. This means that an administrator is a user with all of the same attributes.

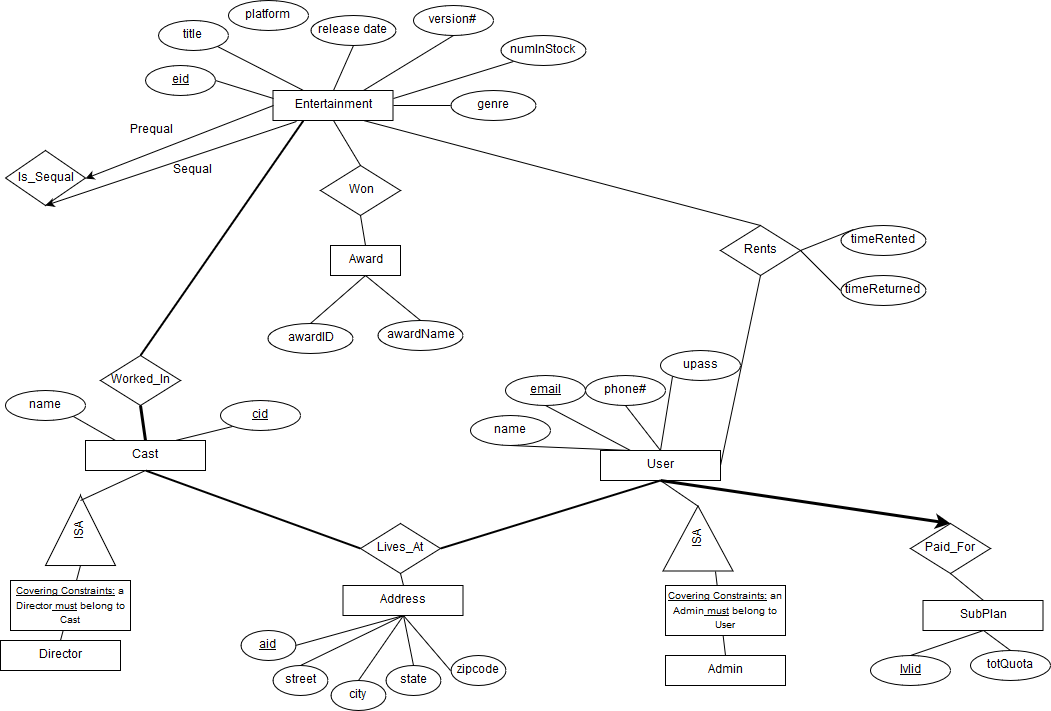
We assumed that each movie can only have one direct sequel. If a user wants to search for all of the sequels, we were planning on designing a query that would collect the sequels in a cascading motion.

For the keyword search, we are only going to be searching for keywords within the title.

Within the ER diagram, we do not have the various permissions represented. For example, we do not have represented that an administrator can add movies or games or that they can change their information. We do, however, have represented that users can rent the forms of entertainment.

For the top 10 forms of entertainment, we assumed that each month starts on the 1st at 12:00am. This means that when the clock rolls over on this day and the database is started up again, the previous month’s information will be wiped, the past month will take its place, and the number of rentals will be set to 0 to start recording for this month. We are also assuming that when we say “past month” we are truly meaning the past month and not the current month. For example, if it is October, we want to see the top 10 rentals for September, not what is top 10 currently for October. Then, when it changes to November, the information for September will be wiped, the information for October will take its place, and we will begin recording information for November.

**ER Diagram**

****

**Relational Schema**

**Entertainment (eid, title, release\_date, genre, num\_in\_stock, sequal\_id, platform, version)**

* “eid” is a Primary Key
* “title” is NOT NULL
* “genre” is NOT NULL
* “num\_in\_stock” is NOT NULL
* “platform” is NOT NULL
* “version” is NOT NULL
* “sequal\_id” Foreign Key to Entertainment

**Rent\_History (rid, eid, uid, time\_rented, time\_returned)**

* “rid” is a Primary Key
* “eid” is a Foreign Key to Entertainment
* “uid” is a Foreign Key to User
* “time\_rented” is NOT NULL
* “time\_returned” is NOT NULL

**Users (email, name, pass, phone, aid, is\_admin, level\_id)**

* “email” is PRIMARY KEY
* “level\_id” is a FOREIGN KEY to Sub\_Plan Table (NOT NULL)
* “aid” is a FOREIGN KEY to Address Table (NOT NULL)
* “name” is NOT NULL
* “pass” is NOT NULL

**Sub\_Plan (level\_id, total\_quota)**

* “level\_id” is a Primary Key
* “total\_quota” is NOT NULL

**Cast (cid, name, aid, is\_director)**

* “cid” is a Primary Key
* “name” is NOT NULL
* “aid” is a Foreign Key to Address Table (NOT NULL)
* “is\_director” is NOT NULL

**Worked\_In (cid, eid)**

* “cid” and “eid” make a composite Primary Key
* “cid” is a Foreign Key to Cast (NOT NULL) (ON DELETE/UPDATE CASCADE)
* “eid” is a Foreign Key to Movies (NOT NULL) (ON DELETE/UPDATE CASCADE)

**Address (aid, state, city, street, zip)**

* “aid” is a Primary Key
* “state” is NOT NULL
* “city” is NOT NULL
* “street” is NOT NULL
* “zip” is NOT NULL

**Awards(awardID, title)**

* “awardID” is a Primary Key
* “title” is NOT NULL

**Won(awardID, eid)**

* “awardID, eid” is a Primary Key
* “awardID” is a Foreign Key to Awards
* “eid” is a Foreign Key to Entertainment