**Project Information**

Introduction:

Have you ever logged onto Netflix and had no idea what to watch next? There’s just so many choices! Choosing what to watch next can be difficult -- but it doesn’t have to be! We’re here to give you the next generation of the online streaming experience with Nextflix! Today, people have so many more important decisions to make, but this shouldn’t be one of them! Nextflix offers a personalized decision making algorithm that will help you find your next favorite show or movie! Nextflix not only keeps track of what the user has watched, but also what shows the user has liked and disliked. Nextflix is then able to find new content for the user to enjoy. Nextflix allows users to maximize their subscription of Netflix by making sure always have something to watch.

Nextflix does this by linking TV shows and movies based on the lead actors/actresses, genre, release date, director, and other similar shows in a database. This database allows the show-finding algorithm to access the next ideal show for the user. With a click of a button the user is off to the next show.

\*\*Nextflix is not responsible for loss of productivity or increased occurrences of “lazy afternoons”. All lost sleep is at the discretion of users.

Requirements:

Admin must be able to add/remove actor, movie, tv show, similar\_to

Amin must be able to remove users

Users must be able to edit their own user data

Users must be able to add new movie, or tv show to has\_watched tables

Users must be able to filter movie/tv show selection

Users must be able to like movies/tv shows in has watched tables

Users must be able to find similar movies/tv shows

Users must be able to create user account

User and admin accounts must have passwords

Application must make sure that SQL attacks are prevented

Users shall not be able to add/remove actor, movie, tv show, similar\_to

Users must be able to add to fan\_of

Users must not be able to edit other users fan\_of or has\_watched entries

Admin must be able to export all tables

Users must be granted admin status from another admin

Users must be able to sort has\_watched movies/tv shows

The user password must be at least 8 characters

The user and admin passwords shall not be the same as their respective username

The admin password must be at least 10 characters

Users must have a valid email

Users must be at least 13 years of age

Users must agree to Nextflix’s terms of use

Users must be able to set parental guidelines on their account

A users’ account must be locked for 6 hours if they fail to log in 6 times

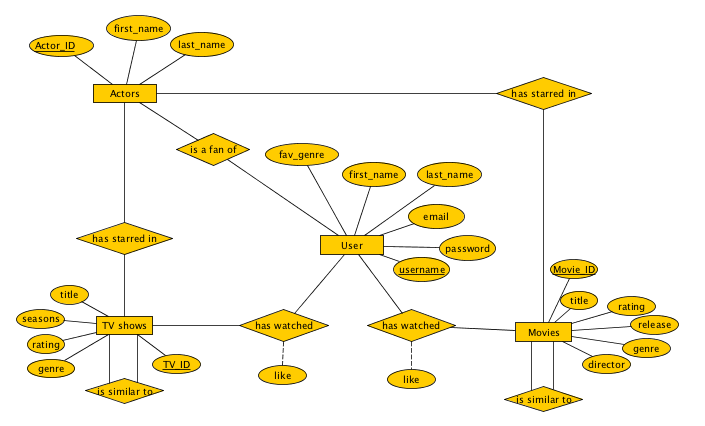
Passwords must be hashed for security purposes

Each user must have a unique email account

**The Design Process**

… Need to build more stuff before we can talk about design decisions

E-R Diagram



Database Schema

...Waiting until database is fully built out

Proof of TNF

User:

A: username

B: password

C: email

D: first\_name

E: last\_name

F: fav\_genre

A->BCDEF

Actors:

G: Actor\_ID

H: first\_name

I: last\_name

G->HI

Movies:

J: Movie\_ID

K: title

L: rating

M: release

N: genre

O: director

J->KLMO

TV shows

P: TV\_ID

Q: title

R: seasons

S: rating

T: genre

P->QRST

Is fan of

A: username

G: Actor\_ID

A->G

Has starred in movie

G: Actor\_ID

J: Movie\_ID

G->J

Has starred in TV

G: Actor\_ID

P: TV\_ID

G->P

Has watched movie

A: username

J: Movie\_ID

U: liked\_movie

A->JU

Has watched TV

A: username

P: TV\_ID

V: liked\_TV

A->PV

Is similar to movie

J: Movie\_ID

W: Movie\_ID2

J->W

Is similar to TV

P: TV\_ID

X: TV\_ID2

P->X

Canonical Cover(Fc)

A->BCDEFGJUPV

G->HIJP

J->KLMOW

P->QRSTX

There are no redundancies or extra relationships to take out since we design the database to only have relevant attributes and relationships. The four main entity sets can be used as candidate keys to access most other relevant data. For example, the username (A) has not only all of their personal information tied to them, but also all of the movies/TV shows they have watched. While having these four big tables is nice because they have all of the relevant information we could need, we are leaving the database as ten tables because most of the information is not necessary for common queries, and it makes the data base easier to manipulate. Each of the 10 tables is in 3NF because they all have one candidate key. Some of them, like is similar to TV could be turned into multivalue tables, but we did not want to have null values because that would waste space and ultimately not be very scalable.

**Evaluation of Product**

...How did we test our application/database

Sample Data and Queries

... SQL code here