**Chris Wong**

**Joshua Frear**

6/11/2013

CS-166

*MovieNet!*

Documentation

## Table of Contents

## Installation

System Requirements

- Linux OS v2.6+
- Intel Xeon Processor (2.33ghz) or better
- 16 MB RAM
- 5 MB HD space

Software Requirements

- Java SE 6
- Postgre SQL

Hence, this will work perfectly for on lab machines!

Step-by-Step Instructions

1. untar the .tgz and place into a directory of your choosing.

2. initialize the postgreSQL database

3.open the 'load_data.sql' file, and replace file paths to your current configeration

4.load the 'create_tables.sql' file

```
-ex "psql -p $PGPORT $DB_NAME < create_tables.sql"
```

5. load the 'load_data.sql ' file

```
-ex "psql -p $PGPORT $DB_NAME < load_data.sql"
```

6.Complie Main.java - ex "javac Main.java"

7. Run the program with the following arguments, <database> <port> <username> <pass>

```
-ex "java Main $DB_NAME $PGPORT wongc password
```

## How to use

Purpose

-Log into your account to order / watch movies!

-You can also comment, rate, or favorite movies!

-In addition, you can track other users and follow their recent activity!

-Superusers, or admins, can add new movies,directors,actors,authors, and delete comments or users!

Controls

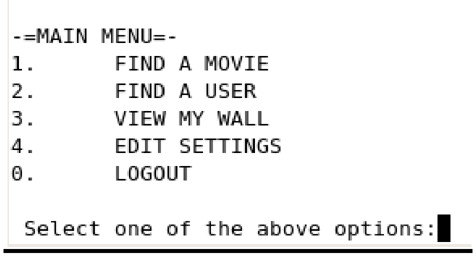
-Simply follow the on screen directions and input your action.

-All inputs will be based number keys

-Press the number key you wish to use, followed by pressing 'enter'

-Generally, inputting '0' in a menu will make the program go back / go to a pervious menu

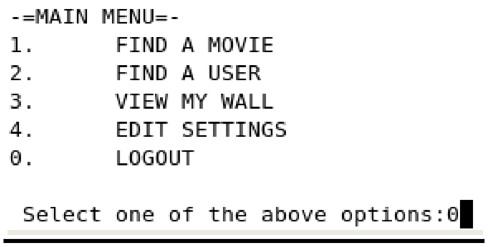
Example



```
-=MAIN MENU=-
1.      FIND A MOVIE
2.      FIND A USER
3.      VIEW MY WALL
4.      EDIT SETTINGS
0.      LOGOUT

 Select one of the above options:
```

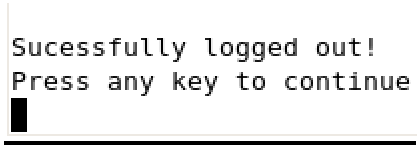
*Lets suppose we want to LOGOUT.*
*Thus, we need to choose the option marked as "0"*
*To do this, we first type 0 into the terminal*



```
-=MAIN MENU=-
1.      FIND A MOVIE
2.      FIND A USER
3.      VIEW MY WALL
4.      EDIT SETTINGS
0.      LOGOUT

 Select one of the above options:0
```

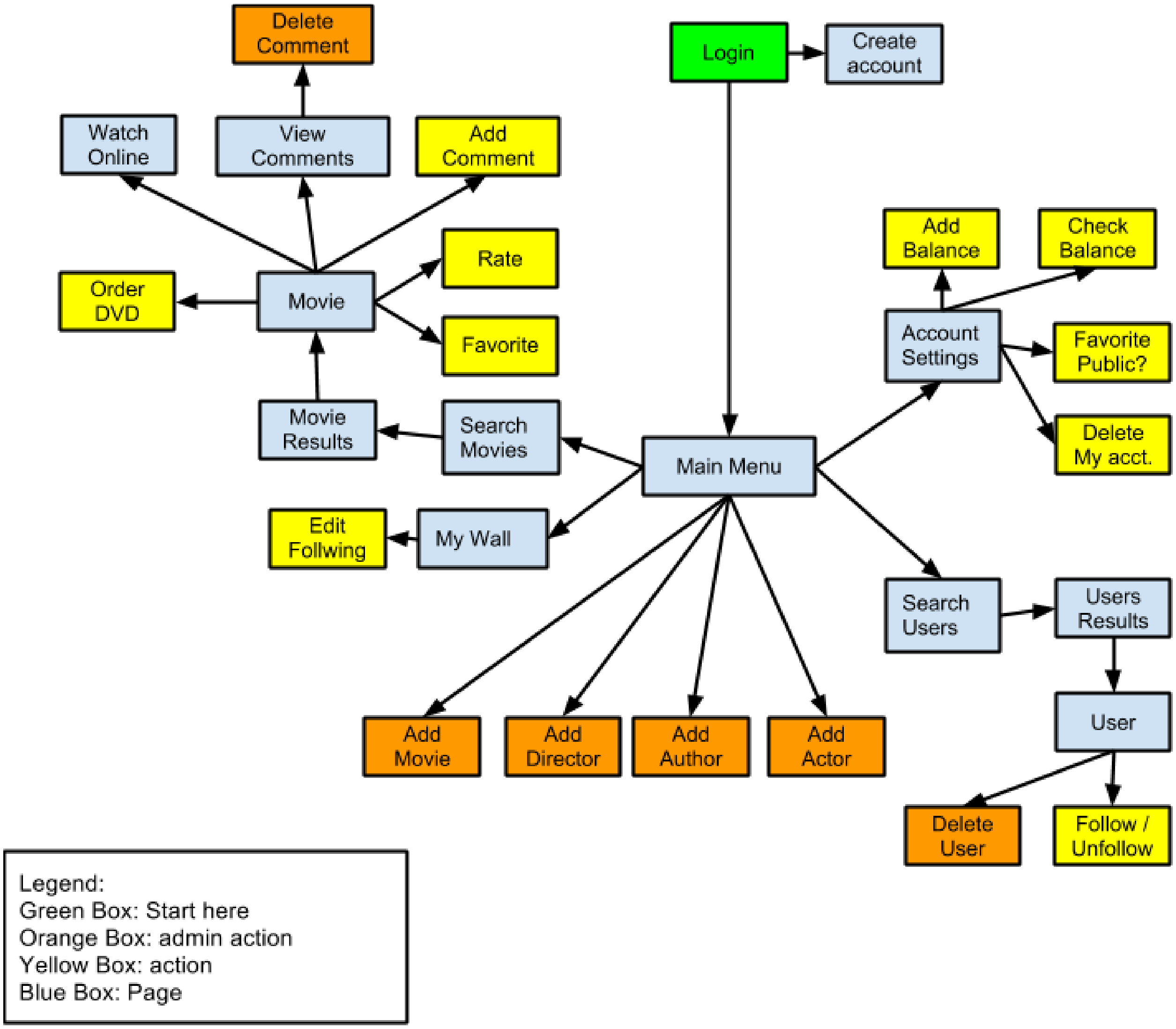
*We have sucessfully logged out!*
*Input any key, such as presssing enter, to continue (to login page)*



```
Sucessfully logged out!
Press any key to continue
```

## Diagram of Program:

Below is a flowchart depicting the implementation of *MovieNet!*



## Important Assumptions

We strictly followed all guidelines posted in phase1.pdf of the project. Thus, here are the main features of the program:

- Online streaming is not actually implemented, only redirects to a dummy page.

- Modifications to create_tables.sql:
  - created timestamp in order & likes to complete time for wall
  - added indexing for extra credit

- Because there is no relation to make a user/profile 'private/public' in the give sql files, we have not implemented "permission on information it shares with it's followers"

- Individual register to use the application, during registration users input
  - Firstname(required), middle name, last name (required), email address(required), stree1,stree2,state,country, and zip code.
- User seraching was not specified, thus we have limited searching to
  - username, email, and list all
- Users has a wall which shows updates on users they follow, and displays:
  - followed user's recently favorited movies
  - followed user's recently rated movies
  - followed user's recent watched movies
  - followed user's recent comments
  - Formate is not specified, thus is organized by user, then update type.

- Movies have the following information:
  - Title, online price, dvd price, genre, category, times watched, rating
  - ratings are integers between 0 and 10
  - when movies are created, they start with an initial rating of 0
  - when movies are created, they start with an initial views of 0
- Moive searching was not specified, thus we limit searching to:
  - title name, director name, selecting a specific genre, and list all

- Superusers can
  - add new movies, directors, authors, or actors
  - delete other users (including other superusers & themselves)
  - delete specific movie comments

- Users can order DVDs to be shipped, or watched online
  - When ordered to be shipped, they do not need to specify an email,address, or username, because that is redundant since this information has already been filled out, and the username would always be the current user.
    - if the email & address has not been filled out, it will prompt them to fill it in.

## Design Choices

- MovieNet! has been implemented entirely in java(1.6) , using a Postgre SQL database
- Two primary files, "Main.java" and "Table.java"
- "Main.java" Contains the bulk of the program,
    - including all individual sql queries, menu creation, menu navagation, and input checking.
- "Table.java" is a simple class, designed to hold query results, and do simple get oprations.
- For simplicity, a console-based GUI was implemented. Simplistic design is easier for users to understand and use.
- Our program is *robust:* All user inputs are checked for validity and program acts accordingly to invalid inputs.

Used github, to see repository go to:
https://github.com/suntinwong/cs166project
Note: user "suntinwong" is Chris Wong

## Extra Credit

- Two indexes are created in create_tables.sql, an index on user_id, and an index on video_id
  - *CREATE UNIQUE INDEX UIndex ON users (user_id);*
  - *CREATE UNIQUE INDEX MIndex ON video (video_id);*
  - user_id and video_id are the most accessed varibles for our queries, since all involve editing either the user or video in some way, we need to get the video id. Thus, we saw it best to create indexes for only these two values.

- Our program is *robust:* All user inputs are checked for validity and program acts accordingly to invalid inputs.

- Minimalistic, simple console-based design.