**CS32 Project 4 Report**

Xuanhe Zhang UID:706078980

You must submit a brief (You're welcome!) report that details the parts of the project:

1. Parts of the project that you were unable to finish.

No.

1. Parts of the project that have bugs that you have not yet been able to fix.

No.

1. Parts of the project that use banned STL components (perhaps because you used them for placeholder implementations but didn’t have time to rewrite the code to remove them)

No.

1. Tests:
   1. I declare a **TreeMultimap<std::string, int> map**.
   2. I use ***insert()*** to insert key-value pairs like “carey” and 0, “carey” and 1, “carey” and 2, “Carey” with 10 to test whether TreeMultimap class work properly.
   3. I use ***find()*** to find “carey” and get its iterator, and print out each value of the key “carey”, check whether they are 0, 1 and 2.
   4. I use ***find()*** to find “smallberg”, which not include in the map, and to check whether the iterator is not valid.
   5. I declare **UserDataBase *udb*** and use **load()** to load “users.txt” file.
   6. I declare **MovieDatabase *mdb*** and use **load()** to load “movies.txt” file.
   7. I use **get\_user\_from\_email()** to get a User from an email “AbFow2483@charter.net

” (first user in data) in “users.txt”.

* 1. Then I call **get\_email(), get\_full\_name(), get\_watch\_history()** to test whether **User** class works properly.
  2. For each movie in the user’s watch history, I call mdb’s **get\_movie\_from\_id()** and print out them to test whether **Movie** calss works properly.

The result is matched with “user.txt”:

*ID22937: Revolution OS*

*ID04742: Spooks Run Wild*

*ID02001: Sorority House Massacre*

*ID05573: Frankie Starlight*

*ID18156: Please Believe Me*

*ID13174: Notorious*

*ID18997: Kano*

*ID17999: Who?*

*ID18925: You're Not You*

*ID25679: Beavis and Butt-Head Do America*

*ID20157: Under the Gun*

*ID09372: Closed For Winter*

*ID25117: The Standoff at Sparrow Creek*

*ID25822: Heavenly Forest*

*ID21918: The Two Firefighters*

*ID17218: 3 A.M*

*ID36917: The Mustang*

*ID11725: How To Frame A Figg*

*ID09830: George Michael: Freedom*

*ID28583: Daddy Long Legs*

*ID07146: Sullivan's Travels*

*ID17151: State of Grace*

*ID00525: Book of Numbers*

*ID11617: Leprechaun: Back 2 Tha Hood*

*ID06425: Saint Jack*

*ID07224: Sole Survivor*

*ID14969: The Taming of the Shrew*

*ID23100: Peter and the Farm*

*ID10786: Project Alien*

*ID15251: The Gay Desperado*

*ID32327: Earthfall*

*ID06853: The Cousin*

*ID34865: Batman: Return of the Caped Crusaders*

*ID09156: Madison Avenue*

*ID30449: The Hurt Business*

*ID26959: Deep West*

*ID07783: America the Beautiful 2: The Thin Commandments*

*ID20249: Tropic Thunder*

*ID35777: The Girl*

*ID15460: The Public Eye*

*ID22199: Genetic Chile*

* 1. I construct a **Recommender** class using ***udb*** and ***mdb***.
  2. I call **recommend\_movies()** to get the user’s recommended movies.
  3. I print some of recommend movies to verify Recommender class works properly.

Here are first five movies that recommended to the user:

*1. Ghosts on the Loose (1943) Rating: 2.16667 Compatibility Score: 132*

*2. 'Neath Brooklyn Bridge (1942) Rating: 3.5 Compatibility Score: 124*

*3. Mr. Wise Guy (1942) Rating: 3 Compatibility Score: 123*

*4. The Death and Life of Bobby Z (2007) Rating: 3.01282 Compatibility Score: 118*

*5. Miller's Crossing (1990) Rating: 3.95467 Compatibility Score: 114*