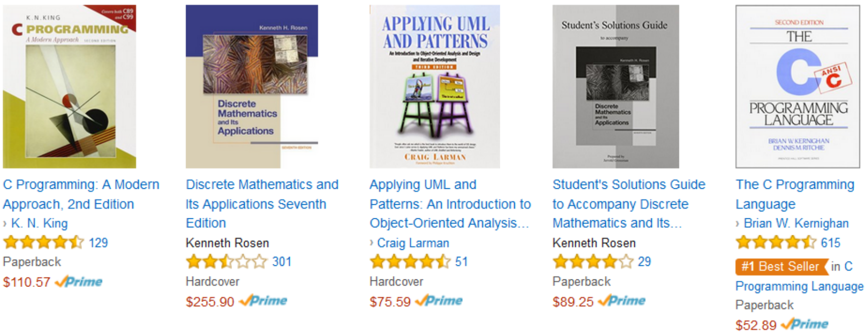
CSC 212 Project
Developing a Ratings Query Application

College of Computer and Information Sciences
King Saud University

Fall 2016

1 Introduction

Most major websites collect feedback from their customers in order to improve the quality of their services. The most common form of user feedback is ratings. For example, Amazon.com collects users' ratings of books (see Figure 1), Imdb.com collects movie ratings and Booking.com gathers users' rating of hotels.



C Programming: A Modern Approach, 2nd Edition
› K. N. King
★★★★½ 129
Paperback
$110.57 ✓Prime

Discrete Mathematics and Its Applications Seventh Edition
Kenneth Rosen
★★½☆☆ 301
Hardcover
$255.90 ✓Prime

Applying UML and Patterns: An Introduction to Object-Oriented Analysis...
› Craig Larman
★★★★½ 51
Hardcover
$75.59 ✓Prime

Student's Solutions Guide to Accompany Discrete Mathematics and Its...
Kenneth Rosen
★★★★☆ 29
Paperback
$89.25 ✓Prime

The C Programming Language
› Brian W. Kernighan
★★★★½ 615
#1 Best Seller in C Programming Language
Paperback
$52.89 ✓Prime

Figure 1: Some books' ratings on Amazon.com

A rating consists mainly of three important pieces of information:

1. The user ID: this is the ID of the user who gave the rating.
2. The item ID: this is the ID of the item being rated. For example, the item can be a book, a movie or a hotel.
3. The rating value: this is a numerical value that is usually an integer ranging from 1 to 5 (or 1 to 10), where 1 is the lowest rating (user not satisfied) and 5 (or 10) is the highest rating (user very satisfied).

Your goal in this project is to write a program that reads a list of ratings from a text file, stores them efficiently in memory (you should choose the appropriate data structure for this) and answers a number of queries as efficiently as possible. Examples of queries are: What is the rating given by user i to item j? What is the average rating of item j? What is (are) the highest rated item(s)?

2 Requirements

In this phase, you are required to implement the following classes:

```
public class Rating {
   private int userId;
   private int itemId;
   private int value; // The value of the rating

   // Constructor
   public Rating(int userId, int itemId, int value);

   // Getters... (No setters. This class is immutable)
}

public class RatingManager {

   // Constructor
   public RatingManager();

   // Read ratings from a file and create a RatingManager object that stores these
       ratings
   public static RatingManager read(String fileName);

   // Add a rating
   public void addRating(Rating rating);

   // Return all ratings given by user i. Search should be efficient.
   public LinkedList<Rating> getUserRatings(int i);

   // Return all ratings given to item j. Search should be efficient.
   public LinkedList<Rating> getItemRatings(int j);

   // Return the average rating of item j. If i has no ratings, -1 is returned
   public double getAverageItemRating(int j);

   // Return the average rating given by user i. If i has no ratings, -1 is
       returned
   public double getAverageUserRating(int i);

   // Return the list of all items having the highest average rating (for example
       if the highest average rating is 4.9, the method should return all items
       with average rating 4.9)
   public LinkedList<Integer> getHighestRatedItems();
}
```

Let n denote the number of users, m the number of items and k the number of ratings. Usually k is much smaller than nm, because a user does not usually rate all items. Your memory requirement must be $O(k)$ and not $O(nm)$.

3 Deliverable and rules

You must deliver:

1. A report written using the provided template.

2. Source code submission to Web-CAT.

The submission **deadline** is: **20/11/2016**.

You have to read and follow the following rules:

1. All data structures used in this project **must be implemented** by the students. The use of Java collections or any other library is strictly forbidden.
2. This project is to be conducted by groups of **three** students. Groups of more than three students are not accepted. Groups of two students are strongly discouraged and can only be accepted with a special permission from the course instructor.
3. All students must **submit** the list of their **group members** within one week of the announcement of this project. Once the groups are chosen, no student can change the group (even if some group members have dropped the course).
4. **Every member** of the group must participate in **all parts of the project**: designing the software, programming and writing the report. Members of the same group may receive different marks according to their participation in the project.
5. The submitted software will be evaluated automatically (using Web-Cat) and in a demonstration (after phase 3) to which all the group members must attend.
6. Any member of the group who fails to **attend the demonstration** without a proper excuse (consult the university and college regulations) shall receive the **mark 0** in the project.
7. In accordance with the university regulation, **cheating** in the project will be sanctioned by the **grade F**.