```java
//author(s): Alex Berliner
package com.cyf.challengeyourfriends;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;

import android.content.Context;
import android.util.Log;

public class ChallengeTopManager {
    //manages the top rated challenge list
    private File fileChallengeList;
    private String challengeListName = "TopChallengeList.txt";
    private Context context;
    private ArrayList<Challenge> challengeList;

    public ChallengeTopManager(Context context, boolean resetFile) {
        this.context = context;
        //init AL
        //
        fileChallengeList = new File(context.getFilesDir(), challengeListName);
//      fileChallengeList = new File("challengeFile.txt");//rp
        printChallengeFile();
        boolean fileIsValid = true;
        //validates file based on structure
        try {
            Scanner sc = new Scanner(fileChallengeList);
            if(!sc.hasNext()){
                fileIsValid = false;
            } else {
                Integer.valueOf(sc.next());
            }
            sc.close();
        } catch (FileNotFoundException e1) {
            fileIsValid = false;
        }
        fileIsValid = !resetFile;
        //reset file if file is invalid
        if(!fileIsValid){
            try {
                fileChallengeList.delete();
                FileOutputStream fos;
                fos = context.openFileOutput(challengeListName,Context.
                    MODE_APPEND);
                fos.write("0".getBytes());
                fos.close();
            } catch (IOException e) {
            }
        }
        challengeList = inflateChallenges();
    }
    //returns an arraylist of challenge names
```

```
    public ArrayList<String> getChallengeNames(){
        ArrayList<String> temp = new ArrayList<String>();
        for(Challenge c : challengeList)
            temp.add(c.name);
        return temp;
    }
    //returns a challenge with the specified name
    public Challenge getChallengeNamed(String str){
        for(Challenge c : challengeList){
            if(str.equals(c.name))
                return c;
        }
        return null;
    }
    //creates the challenge objects from the challenge file
    public ArrayList<Challenge> inflateChallenges(){
        ArrayList<Challenge> AL = new ArrayList<Challenge>();
        if(!fileChallengeList.exists()){
            return AL;
        }
        try {
            Scanner sc = new Scanner(fileChallengeList);
            int numChallenges = Integer.valueOf(sc.nextLine());
            log("numChallenges: " + numChallenges);
            for(int i = 0; i < numChallenges; i++){
                //Name
                String name = sc.nextLine().replaceAll("\n", "");
                //ID
                int ID = Integer.valueOf(sc.nextLine());
                //number of rules to read
                int numRules = Integer.valueOf(sc.nextLine());
                ArrayList<String> rules = new ArrayList<String>();
                for(int j = 0; j < numRules; j++){
                    rules.add(sc.nextLine().replaceAll("\n", ""));
                }
                //number of days left
                int completeDays = Integer.valueOf(sc.nextLine());
                int rating = Integer.valueOf(sc.nextLine());
                AL.add(new Challenge(name, ID, rules, completeDays,rating));
            }
            sc.close();
        } catch (FileNotFoundException e) {
            log(e+"");
            e.printStackTrace();
        }
        return AL;
    }

    public void printChallengeFile(){
        log("#### CHALLENGE CONTENTS START ####");
        try {
            Scanner sc = new Scanner(fileChallengeList);
            while(sc.hasNext()){
                log(sc.nextLine());
            }
        } catch (FileNotFoundException e) {
```

```
        }
        log("#### CHALLENGE CONTENTS END ####");
    }

    public void addChallenge(Challenge challenge) {
        for(Challenge c : challengeList){
            if(c.name.trim().equals( challenge.name.trim())){
                log("Avoiding duplicate entry");
                return;
            }
        }
        challengeList.add(challenge);
        saveChallenges();
    }


    public void saveChallenges() {
        try {
            FileOutputStream fos;
            fileChallengeList.delete();
            fos =
                context.openFileOutput(challengeListName,Context.MODE_APPEND);//
                new FileOutputStream(fileChallengeList, true);//
            fos.write("".getBytes());
            fos.write((challengeList.size()+"\n").getBytes());
            for(Challenge c : challengeList){
                fos.write((c.toString()+"\n").getBytes());
            }
            fos.close();
        } catch (IOException e) {
            log("File write not successful");
            log("error: " + e);
        }
    }


    public void wipeChallenges() {
        try {
            fileChallengeList.delete();
            fileChallengeList.createNewFile();
        } catch (IOException e) {
            log("Friends list create failed");
            e.printStackTrace();
        }
    }
    void log(String str) {
        Log.w("Challenges", str);
//      System.out.println(str);
    }

    public ArrayList<Challenge> getChallenges() {
        return challengeList;
    }

    public void removeChallenge(String challengeName) {
        for(Challenge c : challengeList){
```

```
            if(c.name.trim().equals(challengeName.trim())){
                challengeList.remove(c);
                log("Removing entry " + challengeName);
                break;
            }
        }
        saveChallenges();
    }
}
```