EnduRan
A Web Application for Managing Racing Events

Jigarkumar Patel, Sergiu M. Dascalu
Department of Computer Science & Engineering, University of Nevada Reno

Abstract

Organizing a racing event is not an easy task. Organizing a racing event that spans across multiple days and whose participants have to run (on foot) for 100 miles is even harder. In such events many checkpoints are needed to provide runners with access to food, water and, if necessary, medical assistance. At these checkpoints updates on participants' health status and race timings need to be sent to a central location so that followers across the globe can monitor the overall progress of the race as well as the current situation of individual racers. This paper examines development challenges and presents a flexible open source solution for a web-based software application, EnduRan,z that we created for managing long-distance racing events. The paper takes a closer look at the need for this software application and gives details on the application's specification, design, and implementation. Design guidelines resulted from this software project, together with several lessons learned from it, are described as well in the paper. Results of using EnduRan in the 2009 and 2010 Lake Tahoe 100 Miles Endurance Races and pointers to future work are also included.

1. Introduction

Organizing and managing a racing event is a complex task. Monitoring closely such an event and updating the live status of the race makes the challenge even more difficult.

Most of the software solutions available on the market are desktop-based applications [1, 2, 3, 4, 5]. These applications are usually dependent on specific operating systems and have different system requirements for installation. Furthermore, they are generally rigid in terms of structure and functionality, and usually lack well defined operational workflows. Importantly, these applications cannot be used via remote clients. Indeed, there are very few software applications available that can manage races such as foot races, marathons, bicycling events, and swimming events. In addition, to the best of our knowledge, there is only one application that offers a web-based solution [6]. With it, however, the organizers must register and manage the events using the company's web server.

All the applications mentioned above are commercial and rather expensive for the organizers of racing events, many of them non-profit organizations. In contrast, our proposed solution, EnduRan, is web-based and freely available for everyone interested to use it. In addition, the solution is flexible such that it can be used to organize various types of races, including running races, bicycle races, swimming races, and endurance triathlons.

The project was initially designed for the Tahoe Mountain Milers Running Club (TMMRC) [7]. A non-profit organization, the TMMRC is a member of the Road Runner Club of America and the Pacific Association of USA Track & Field. The club provides a wide range of running opportunities for residents and visitors of Lake Tahoe and the areas around it (extending in both Nevada and California). The club has organized various racing events since 2002.

The TMMRC's main event, typically held in July, registers several hundreds of participants competing in three race categories: 50 kilometers (50K), 50 miles (50M), and 100 miles (100M). The TMMRC club came up with the proposal for the present web-based software project to the Department of Computer Science and Engineering of University of Nevada, Reno, where the authors of this paper work. Thus, from its inception the main motivation behind the EnduRan project has been to help a non-charitable organization accomplish their worthwhile cause. While working on the project, we realized that the research, design, and development efforts invested in it could offer some useful experiences as well as supporting software tools to other interested race event organizers, software engineers, and the general public (in particular, runners and other athletes). In this paper, we describe technical details of this software project and briefly summarize our development experiences.

This paper, in its remaining part, is organized as follows: Section 2 describes several challenges related to managing racing events and overviews the problems pertaining to using the TMMRC's previous race management approach. The proposed EnduRan web-based solution is described in Section 3 in terms of features, architectural design, site map, and technologies

978-0-7695-4367-3/11 $26.00 © 2011 IEEE
DOI 10.1109/ITNG.2011.18

used for its implementation. Results of using EnduRan in real racing events, the 2009 and 2010 Lake Tahoe 100 Miles Race, design guidelines, and lessons learned are presented in Section 4. Section 5 discusses future improvements and features that can be added to make the EnduRan application more versatile and robust. Finally, Section 6 concludes the paper by summarizing the contributions of our work.

2. Problem

2.1 Problem background

The Tahoe Mountain Milers Running Club (TMMRC) was founded in 1995. The club provides many running opportunities for visitors and residents of the Reno-Tahoe area. The TMMRC main event, a set of running races in Lake Tahoe's vicinity usually attracts between 400 and 500 participants competing in various race categories: 50K, 50M, and 100M. Notably, the lake's altitude is about 6,225 feet (about 1,870 meters). These races, held concurrently, are very long and have to include several checkpoints for the participants. Thus, the club sets up various such checkpoints for food, water, and medical assistance.

The most important task of the checkpoint staff is to log race progress by taking the participants' arrival times at each checkpoint. Before using the EnduRan web application presented in this paper, the check-in process at each checkpoint could take up to one minute for each participant (we note here that as a non-profit organization TMMRC did not have so far resources to use electronic tags attached to participants and automatic scanning for check-in, a situation that will likely change in the future, most probably from 2011 on – for more details please see Section 5). This processing time used to add up quite significantly to the race management overhead as there are several checkpoints (between 6 and 16) for each race length and several hundreds of participants. Furthermore, compiling check-in data used to take even longer. Also, all the data used to be logged on paper by checkpoint officials. As it can be seen, this manual check-in process as well as the location of checkpoints (up in the mountains, at altitudes above 1800 meters) made it practically impossible to report to the followers the progress of the overall race and of individual runners (the followers include the participants' families and friends, as well as the general public).

2.2 Generic race process

Usually an endurance race has one or more checkpoints as part of the race. In practice, for the 100M race there are 16 checkpoints and some of them share the same physical location (e.g., the checkpoint at mile 11 and at mile 61 can be in the same place). The race begins at a pre-defined start point and finishes at a specified end point, which may or may not be the same with the start point. Each participant starts within the time frame assigned to the participant's group (e.g., 50K, male, group age 31-40). Each participant must visit and check in at each checkpoint, including the end point, in order to complete the race. At each checkpoint he or she is provided with food, water, and medical assistance if needed. Race officials log in the participants' check-in time at each checkpoint. At the end of the race, organizers compile the data collected at all checkpoints for all participants. Using this data the officials can determine the order in which the race was completed by the participants. As the organizers are holding multiple race categories on the same track, they have to repeat this process for each race.

2.2 Issues with the previous race management approach

Until July 2009 TMMRC used to manage its races using pen and paper. Several weeks before the race started the organizers (all volunteers) worked on collecting the participants' information via mail. As TMMRC organizes more than one race at a time, the organizers had to sort, verify and assign individual numbers (bib numbers) to all participants. As detailed before, during the event, which took more than two days, the officials worked hard to log on paper each participant's check-in time at each checkpoint. At the end of the race the organizers collected all the data and compiled the data manually to obtain the final results of the race.

This approach has worked in the past but has several significant problems. First of all, every step of the management process is very time consuming. It takes numerous hours of work to collect and compile all the data. Moreover, there is no transparency in the process and there is no way that a participant or a follower can get current information on the status of the event. Also, because all the data was collected using pen and paper the process was prone to human errors. Overall, by using the "manual" approach it was really hard for the organizers to manage reliably the entire race organization and monitoring process. Therefore a new, more effective, software-supported solution for race management was necessary.

3. The proposed solution

Given the above difficulties with managing racing events by TMMRC, providing a new, flexible, effective computer-based solution that encapsulates the entire management process under one hood has been the objective of EnduRan. In the following subsections the proposed solution, designed, implemented, tested, and applied in practice in 2009 and 2010 is described briefly in terms of main features, high-level architecture, data design, site map, and technologies used. These are only some of the more salient details of the software engineering documentation [8] generated in the project that we undertook to create EnduRan.

3.1 *EnduRan* features

Among EnduRan's more important required features are the following:

Feature 1: Allow the organizers to manage a racing event. This means that the organizers can create, edit, and delete a racing event. This also allows organizers to associate participant and checkpoint information to a specific race.

Feature 2: Allow the organizers to manage checkpoints. This feature makes it possible to create, edit, and delete checkpoints. At the time of its creation each checkpoint is assigned to a single race. The organizers can also store general information about items such as distance from/to given locations and services available at the checkpoint.

Feature 3: Enable the organizers to manage race participants. This allows the organizers to add, edit, and delete participant information. Each participant can be associated with a single race only.

Feature 4: Support an automated check-in process (note that this feature will only be implemented for the 2011 version of the 100M race). Before the beginning of the race each participant will be handed over his or her Radio-Frequency Identification (RFID) badge [9]. The participant will be able to use the given badge at each checkpoint to check-in. He or she will just need to pass his or her given badge over the RFID reader placed at each checkpoint.

Feature 5: Enable live race progress presentation on the web. This feature allows race followers to access information about the current state of the race as well as about the situation of individual participants. This feature has to use previously collected data at checkpoints to generate various live race reports.

Feature 6: Generate end race reports. This capability allows the organizers to generate at the end of the race various presentations of the results, including statistical reports and graphs.

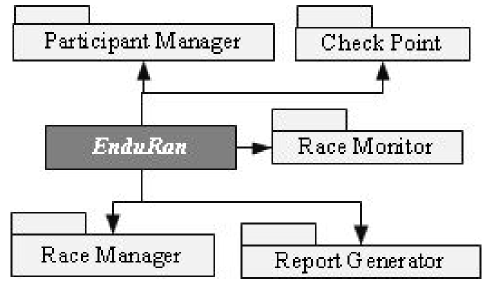

Figure 1: EnduRan System Level Diagram

Feature 7: Use minimal bandwidth. As the race takes place up in the mountains where there is only poor, sporadic cell phone communication, EnduRan needs to use only a minimal number of requests to the central server, and transfer with it only a small amount of data.

3.2 High-level design

Based on the above set of features, defined as key requirements for EnduRan, the system's resulting architectural design diagram [8] is shown in Figure 1. Each subsystem contains a set of functions that implement specific features of the system. The subsystems are independent from an operational point of view but are interconnected through data communication and data dependency. Each subsystem is further described as follows:

Race Manager: This subsystem allows the organizers to create and manage racing events. It allows the organizers to enter the race length, location, description, time, and other related details. It also stores the URL of the race map. Race organizers can also modify (edit) or delete a racing event. This subsystem provides the necessary input (a new race event) for the entire system. From this point of view, all other subsystems depend on this subsystem.
Participant Manager: The main users for this subsystem are again the race organizers. It allows them to add, modify, and delete participants for a particular racing event. It also takes care of displaying participant information to the public once participant data is available and released for publication. This subsystem depends on the Race Manager subsystem in that it needs an existing racing event to which participants can be associated.

Race Monitor: This subsystem enables the public's open access to race information. Race followers can monitor the evolution of individual participants as well as the progress of the race as a whole. For example, the public can view the race's current leaders as well as checkpoint-related information. It is important that this module is optimized in order to handle massive usage loads on the server.

Check Point: This is the subsystem that manages all checkpoint-related activities. In essence, it allows the race organizers to create, modify, or delete a checkpoint. Each checkpoint is associated with a single race event. The subsystem also incorporates the participant's check-in and check-out process at the checkpoint. The race officials can monitor the check-in process and, as needed, can also add specific comments for individual participants (for example, a caution message in case the physical condition of a participant needs to be checked more thoroughly at the next checkpoint).

Report Generator: This subsystem is used to create reports. Some of the reports are available to the public but most of them can be accessed by race organizers only. The subsystem can generate various reports based upon given conditions (more details are available in Section IV, Results and Discussion).

3.3 Data design

The EnduRan solution is highly dependent on its stored data. Figure 2 contains the diagram of the system's main database entity-relationship model. The Race table is required by all other tables.

3.4 Site map

The EnduRan software is a web-based application. Its functionality, as seen by its users, is represented using a site map. The application's externally visible functionality matches its internal high-level system design.

3.5 Technologies used

In terms of technologies, EnduRan uses jQuery [10] as a JavaScript[11] framework, PHP [12] as a server side scripting language, MySQL [13] as backend database server, Apache as web server [14], Linux[15] as a server operating system, and freely available web templates for the user interface layouts. The application also uses Google Maps [16] to display various race track maps. Further, it also utilizes jQuery UI library for CSS framework and controls [10].

4. Result and discussion

The operational EnduRan web application was deployed in July 2009 on the newly registered domain www.trtlive.com [17]. At that time, TMMRC held three races concurrently on the same track around Lake Tahoe. EnduRan was thoroughly tested in the field, during more than two days on the 100 miles race, which had 131 participants and 16 checkpoints in total. Some results

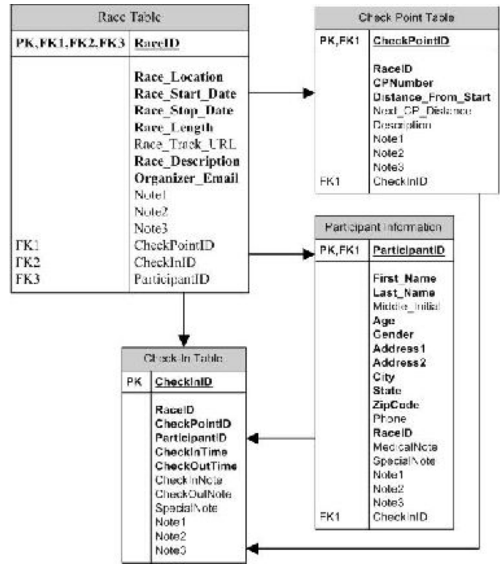

Figure 2: EnduRan Database ER Model

generated during the event and presented to the public are still available at [17]. The application worked very well, and performed virtually error-free under a workload of more than 20,000 hits over about 56 hours of operation in the 2009 event and over 40,000 hits during the 2010 event. (there were several human operator errors due to limited time for training, but no software execution errors).

In the following, couple of selected snapshots illustrating the EnduRan application "in action" is briefly described. Due to space limitations, they represent only a fraction of the application's full set of capabilities; thus, for more details the interested reader is referred to either [17] or [18,19]. The latter includes the application downloadable code, a larger selection of interface snapshots, and several photos from the 100M race itself. The initial interface for managing racing events is shown in Figure 3. Note that the different colors of the background (orange, yellow and white) denote, respectively, past, current and future races. Figure 4 presents an excerpt of the end race results – note that in round rackets are the actual times taken by participants to reach a checkpoint from the start point.

The implemented EnduRan web application has many advantages over the previous approach used by TMMRC [7]. EnduRan offers a flexible, fast, and reusable solution that operates well under restricted conditions (these conditions are derived largely from the specific nature of the problem it solves and the high altitude environment in which the application is used). The distinguishing characteristics of this application, which can also serve as research and design guidelines for similar projects, are as follows:

Figure 3: Interface for Mapping Racing Events

- EnduRan is user centric [21]. It is easy to learn and use, its interface being clear and simple while still attractive for enabling a positive user experience. The application can also be run on mobile phones;
- EnduRan was designed for minimal bandwidth usage. Due to the environment in which is typically used (long endurance trail races at high altitudes), we employed several techniques to minimize the bandwidth used: partial page update via Ajax implementation [21], small communication request and response data size, advanced caching technique deployed for the results page [22], a single image for CSS to reduce the number of web server "get" requests, and web-server enabled to deliver content in compressed format.
- EnduRan takes advantage of rapid code development and simple data management. The former is made possible by using jQuery and jQuery-UI standard JavaScript libraries [10] to speed up the development process and minimize the implementation and updating (maintaining) the code. The latter is acquired through efficient storage, fast retrieval, and streamlined interface for data management.

In addition to the above guidelines resulted from working on the project, we note several other learning experiences. For example, in EnduRan we used the Model View Controller (MVC) based architecture. The MVC-based design allows enhanced software layer abstraction, better code writing management, and easier program maintenance. On top of the solutions described above, for future developments we suggest to employ "in-browser" data storage for offline situations. If a given checkpoint loses its Internet connection due to weak wireless signal reception in a remote location, it will still allow the participants to check in. In such conditions, JavaScript code loaded in the browser will store the data in the memory. When the Internet connection is re-established the same script code will submit all the pending check-in transactions. This will make EnduRan more robust even in remote operating locations.

There are only few commercial or freely available applications for racing event management needs addressed in this paper. Unlike most of the existing desktop-based commercial applications designed for similar purposes, EnduRan is a freely available web-based application that can also be utilized as a desktop application, with a single entry point for the data. While currently operational and reliable, EnduRan can still benefit from a number of further improvements, as outlined next.

5. Future Work

We plan to enhance this web-based application with the following additional features and capabilities. First, the application will be updated to handle automatic scanning of the participants' RFID badges. This will likely be changed soon, and will result in the elimination of manual checking-in of participants at checkpoints. Second, the online registration of participants also needs to be completed. We have not focused on it so far because the main priority was on the reliable management of the event during the race. Third, for easier deployment and application to other types of racing events, a "one click" installation needs to be made available. This will save time to the end user by eliminating the need of creating new database tables and speeding up the application's installation. Fourth, a facility to show track maps with checkpoints pinned on the maps will also be useful. Fifth, a comprehensive set of visual representations of both partial and final race results will make EnduRan more appealing. Such representations could include a variety of graphs for displaying the actual progress of the race as well as predictions on the overall race and the performance of individual participants. Yet another direction of future work could be to create a generic core framework for managing race events. This will allow higher levels of customization while maintaining the same main set of functional capabilities and the same database resources and data access mechanisms.

6. Conclusions

The EnduRan web application for managing racing events started from our desire to support a non-profit organization achieve its worthwhile goal of organizing on a volunteer basis long endurance running races. By relying on a set of appropriate design and implementation decisions, the application we have created was able to operate reliably in real world conditions and support adequately the organization and management of the 2009 Lake Tahoe 100 Miles Endurance Race and three concurrent races during the 2010 event.

The main contributions of the work presented in this paper are as follows. First, EnduRan offers an example of web application aimed at operating under limited digital communication conditions (due to the high altitude, mountainous type of location where the races take place). Besides employing several technical solutions to answer the need for minimal bandwidth usage, the application also addresses the needs for simple and efficient user interface design, simple database management, and rapid code development and evolution. This characteristic is useful because in contrast with existing desktop-based applications it allows simultaneous, remote usage by several categories of users, including administrators (usually, working at a central place), race officials (at checkpoints), and the general public (anywhere in the world, using either regular computers or portable devices such as cell phones.

Second, EnduRan is available to all interested users. This characteristic comes with the all benefits of open source projects, including no financial cost involved and progressive evolution through feedback and participation from the community of users and developers.

Third, EnduRan answers a real, practical need, and while designed for TMMRC's long endurance running races it can also be used for a variety of other similar events, including bicycling races, swimming races, and triathlon events.

Acknowledgements

The authors would like to thank Mr. George Ruiz from TMMRC for initiating this project, providing feedback on its progress, and volunteering his time for organizing the Lake Tahoe long distance endurance races.

References

[1] Race Management Systems, Software for Race Management, http://www.theracedirector.com/, accessed Sep 14, 2010.

[2] Series Tracker, Motocross Race Management Software, http://www.seriestracker.com/st_features.asp, Sep. 14, 2010.

[3] Race Forever, http://www.raceforever.com/, accessed Sep. 14, 2010.

[4] Derby Master Race Management Software, http://www.enterprisingideas.com/derbymaster/, accessed Aug. 25, 2010.

[5] Road Race Management Software, Software for Road Race and Cross Country Meet Management, http://www.raceberryjam.com/roadsoft.html, accessed Sep. 25, 2010.

[6] Race Management System – Features of Online Race Registration, Active Endurance, http://www.activeendurance.com/features.htm, Sep. 11, 2010.

[7] Tahoe Mountain Milers & Sagebrush Stompers Running Clubs, http://tahoemtnmilers.org/, accessed Aug. 25, 2010.

[8] Ian Sommerville, *Software Engineering*, 8th ed., Addison-Wesley, 2006.

[9] Roy Want, "The Magic of RFID", *ACM Queue*, vol. 2, no. 7, pp. 40-48, October 2004.

[10] jQuery, The Write Less Do More JavaScript Library, http://jquery.com/, accessed Sep. 23, 2010.

[11] JavaScript Tutorial, http://www.w3schools.com/JS/default.asp, accessed Sep. 23, 2010.

[12] PHP Language Ref. Manual, http://www.php.net/manual/en/langref.php, accessed Sep. 23, 2010.

[13] MySQL Developer zone, http://dev.mysql.com, accessed Sep. 23, 2010.

[14] The Apache HTTP Server Project, http://httpd.apache.org , accessed Sep. 14, 2010.

[15] The Linux Home Page at Linux Online, http://www.linux.org/, accessed Aug. 8, 2010.

[16] Google Maps Help, http://maps.google.com/support/?hl=en, accessed Aug. 25, 2010.

[17] Tahoe Mountain Milers, Race Results, www.trtlive.com, accessed Aug. 8, 2010.

[18] Jigar Patel's home page, EnduRan section, http://www.siwts.com/enduran.html, accessed Jan. 8, 2010.

[19] Jigar Patel, EnduRan: A Web Application for Managing Racing Events, Master Thesis, Department of Computer Science & Engineering, University of Nevada Reno, 2010

[20] Stephen Heim, *The Resonant Interface: HCI Foundations for Interaction Design*, Addison-Wesley, 2007.

[21] Arno Puder, "A cross-language framework for developing AJAX applications", *Proceedings of the 5th Intl. Symposium on Principles and Practices of Programming in Java*, vol. 272, pp. 115-112, 2007.

[22] Ming-Kuan Liu and Fei-Yue Wang, "Web caching: a way to improve web QoS", Journal of Computer Science and Technology, vol. 19, no. 2, pp. 113-127, March 2004.

Race Result

Bib#	Full Name	Hobart Road I	Tunnel Creek I	Tunnel Creek / Red House I	Mount Rose - I	Tunnel Creek / Mount Rose I	Hobart Road - II	Snow Valley - I	50 Miles	Hobart Road - III	Tunnel Creek II	Tunnel Creek / Red House II	Mount Rose - II	Tunnel Creek / Mount Rose - II	Hobart Road - IV	Snow Valley - II	End Of The Race
553	Erik Skaden		07:01:04 (02:01:04)	10:54:32 (05:54:32)	09:32:55 (04:32:55)	16:21:36 (11:21:36)			13:45:00 (08:45:00)		16:36:41 (11:36:41)	18:00:31 (13:00:31)	20:02:13 (15:02:13)	22:01:56 (17:01:56)			01:27:32 (20:27:32)
561	Robert Evans		07:01:45 (02:01:45)	08:08:43 (03:08:43)	09:41:47 (04:41:47)	11:11:10 (06:11:10)			14:07:00 (09:07:00)		16:48:39 (11:48:39)	18:21:18 (13:21:18)	20:17:37 (15:17:37)	22:09:35 (17:09:35)			01:46:54 (20:46:54)
529	Brett Rivers		08:11:54 (03:11:54)	11:29:00 (06:29:00)	09:50:14 (04:50:14)	17:19:44 (12:19:44)			14:27:00 (09:27:00)		18:44:31 (13:44:31)	22:27:51 (17:27:51)	20:37:50 (15:37:50)				01:50:40 (20:50:40)
518	Ian Torrence		06:58:10 (01:58:10)	11:00:01 (06:00:01)	09:34:55 (04:34:55)	16:48:49 (11:48:49)			13:55:00 (08:55:00)		17:02:14 (12:02:14)	18:44:23 (13:44:23)	20:41:19 (15:41:19)	22:36:51 (17:36:51)			03:02:45 (22:02:45)
508	Pierre-Yves Couteau		08:32:05 (03:32:05)	11:52:44 (06:52:44)	10:14:54 (05:14:54)	17:47:02 (12:47:02)			14:57:00 (09:57:00)		23:33:16 (10:33:16)		21:39:06 (16:39:06)				03:12:59 (22:12:59)
533	Bob Shebest		06:57:50 (01:57:50)	08:05:31 (03:05:31)	09:33:09 (04:33:09)	10:54:52 (05:54:52)			13:57:00 (08:57:00)		16:37:04 (11:37:04)	23:57:55 (18:57:55)	22:18:56 (17:18:56)	00:17:11 (19:17:11)			03:44:06 (22:44:06)
524	Bret Sarnquist		07:22:47 (02:22:47)	08:31:59 (03:31:59)	10:10:14 (05:10:14)	11:44:20 (06:44:20)			15:01:00 (10:01:00)		18:12:25 (13:12:25)	19:39:46 (14:39:46)	21:55:01 (16:55:01)	00:04:30 (19:04:30)			04:08:24 (23:08:24)
571	Sean Lang		07:01:14 (02:01:14)	11:38:18 (06:38:18)	09:57:36 (04:57:36)	17:46:52 (12:46:52)			15:00:00 (10:00:00)		19:23:31 (14:23:31)	23:43:12 (18:43:12)	21:39:33 (16:39:33)				04:17:38 (23:17:38)
507	Ben Doar		07:12:33 (02:12:33)	08:29:15 (03:29:15)	10:16:23 (05:16:23)	12:02:34 (07:02:34)			15:08:00 (10:08:00)		18:43:56 (13:43:56)	20:15:18 (15:15:18)	22:28:50 (17:28:50)	00:29:41 (19:29:41)			04:25:00 (23:25:00)
510	Ben Crew		07:38:56 (02:38:56)	08:54:43 (03:54:43)	10:39:06 (05:39:06)	12:23:23 (07:23:23)			15:40:00 (10:40:00)		18:43:35 (13:43:35)	20:18:20 (15:18:20)	22:27:32 (17:27:32)	00:27:25 (19:27:25)			04:28:58 (23:28:58)

Figure 4: Sample End Race Report