

조정 경기 기록 관리를 위한 모바일 애플리케이션 구현*

김선경, 이준선, 박창욱, 백세인, 이용규
동국대학교 컴퓨터공학과-서울
e-mail: ovvlvlbm@dongguk.edu

Implementation of a Mobile Application for Rowing Race Scoring Management

Seon Kyung Kim, Jun Seon Lee, Chang Wook Park,
Se In Baek, Yong Kyu Lee

Department of Computer Science and Engineering, Dongguk University-Seoul

요 약

국내에서 열리는 조정 경기에 대한 관심이 높아짐에 따라 경기 기록의 체계적인 측정과 관리의 필요성이 대두되고 있다. 본 논문에서는 조정 경기 기록을 측정하고, 경기 정보와 측정 기록을 확인하고 관리할 수 있는 조정 경기 기록 측정 및 관리 모바일 애플리케이션을 구현하고자 한다. 경기장 내 구간 별로 애플리케이션을 사용해 각 심판이 조정 경기 기록을 측정하고, 측정된 기록 정보를 서버의 데이터베이스에 저장한다. 데이터베이스는 데이터 이식이 쉬우며 가상화 서버에 적합한 MariaDB를 사용한다. 경기 운영 본부와 심판은 웹이나 애플리케이션을 통해 데이터베이스에 저장된 경기 정보 확인이 가능하다. 또한 경기장의 각 구간에서의 경기 상황과 경기 기록을 공유하여 국내 조정 경기의 원활한 운영과 기록 관리에 도움이 될 것으로 기대한다.

1. 서론

현재 국내에서 열리는 조정 경기의 기록 측정과 기록 관리는 주 심판이 타이머를 이용하여 직접 기록을 측정하고 보조 심판이 기록을 받아 적은 뒤, 관리자가 구간 별 경기 기록을 취합하여 기록관리 프로그램에 입력하는 방식으로 이루어진다[1]. 그러나 관리자가 기록관리 프로그램에 경기 정보와 함께 직접 기록을 입력해야 하는 불편함이 있어 기록 측정 및 기록 관리 등이 효율적이지 못한 실정이다.

국외에서 열리는 조정 경기의 경우 경기장의 구간 별로 스마트 게이트웨이 혹은 고속 촬영 카메라와 같은 최첨단 장비를 설치하여 기록을 측정한다. 측정된 기록을 실시간으로 이미지 프로세싱 함으로써 경기 기록 측정의 정확성을 높이고 경기를 효율적으로 운영하고 있다[2]. 이에 따라 국내에서도 조정 경기의 정확한 기록 측정과 관리, 데이터 분석 등을 위한 효율적인 시스템의 필요성이 대두되고 있다.

국제적인 표준에 맞춘 조정 경기장의 길이는 총 2,000m로 스타트 라인과 피니시 라인 이외에도 각 500m, 1,000m, 1,500m 구간마다 구간 기록을 측정한다[3]. 세 개의 구간 기록 측정 지점과 피니시 라인에 있는 심판들은 스타트 라인으로부터 해당 구간까지의 경기 기록과 순위

를 측정한다. 국내에서 사용하는 방식으로 기록을 측정하기 위해서는 두 명의 심판이 필요하다. 주 심판이 타이머를 이용하여 기록을 측정하고, 보조 심판이 측정된 기록을 수기로 받아 적는다. 경기가 종료된 후 관리자가 구간 별로 측정된 경기 기록과 최종 경기 기록을 취합하여 기록관리 프로그램에 직접 데이터를 입력한다[1].

현재 대한조정협회에서 사용하는 기록관리 프로그램은 해당 시스템이 설치되어 있는 저장 장치를 기반으로 한 프로그램이다. 국내에서 진행된 경기와 국내 선수들이 출전한 경기에 대한 정보와 경기 기록을 저장하기 위한 용도로 사용된다. 경기 기록이 아닌 경기 운영을 위한 정보는 별도로 관리자가 문서를 작성하여 관리한다[1].

기록 관리에 사용되는 데이터베이스로는 Microsoft SQL Server, Oracle, MySQL, MariaDB 등을 활용하고 있다[4]. Microsoft SQL Server는 트랜잭션과 질차에 대해 높은 수준의 제어를 제공하나 애플리케이션 개발보다는 워크로드를 처리하는데 적합하다. Oracle은 대량의 데이터를 처리하는데 적합한 상업용 데이터베이스로 다양한 운영체제에서 사용 가능하다. 그러나 데이터 안정성을 위한 작업이 많아 데이터 삽입이나 수정이 잦은 시스템에는 적합하지 않다[4]. 또한 MySQL은 설계가 간단하고 호환성이 높지만 더 이상 오픈소스가 아니라는 단점이 있다. MariaDB는 MySQL과 동일한 구조를 가지고 있는 오픈소스 기반의 DBMS이므로 가상화 서버에 적합하다[5].

* 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음(2016-0-00017)

본 논문에서는 MariaDB를 이용하여 조정 경기 데이터 관리가 가능한 애플리케이션을 구현하고자 한다. 조정 경기 기록을 측정하고 경기 기록과 정보를 MariaDB에 데이터베이스에 저장한 뒤, 저장한 데이터를 효율적으로 관리한다. 기록 측정부터 기록 관리에 이르기까지의 일련의 과정을 통합시킬 수 있으며, 효율적인 관리가 가능할 것으로 기대한다.

2. 관련 연구

현재 국내에서 열리는 조정 경기의 기록 측정은 각 구간마다 기록 측정을 위해 대기하는 두 명의 심판에 의해 이루어진다. 두 명의 심판은 주 심판과 보조 심판으로 구성되어 있다. 주 심판이 타이머를 이용해 기록과 순위를 측정하면 보조 심판이 수기로 받아 적는다. 주 심판은 육안으로 보트가 해당 지점을 통과하는 시점을 확인한 뒤, 타이머로 기록을 측정하고 통과한 순서대로 보트의 숫자를 호명한다. 보조 심판은 순서대로 호명된 보트와 기록을 수기로 받아 적는다. 경기가 종료된 뒤, 관리자는 구간별로 측정된 구간 기록과 최종 기록을 취합하여 정리하고 기록 측정 시스템에 직접 정보를 입력한다[1].

국외에서는 기록경기를 위하여 Photo-Finish Cameras와 Fully Automatic Race Timing System[2]을 사용하고 있다. Fully Automatic Race Timing System은 경기 중 기록 측정이 필요한 위치에 Photo-Finish Cameras를 설치하여 기록을 측정한다. 여러 각도에서 피니시 라인을 캡처하여 모호한 상황에서도 정확하게 기록을 측정하고 우승자를 가려내는데 그 목적이 있다[2]. 기상 상태가 경기 결과에 영향을 주지 않고, 동일한 조건에서 이루어지는 기록경기에 적합한 시스템으로 육상, 수영 등의 기록경기에 사용된다.

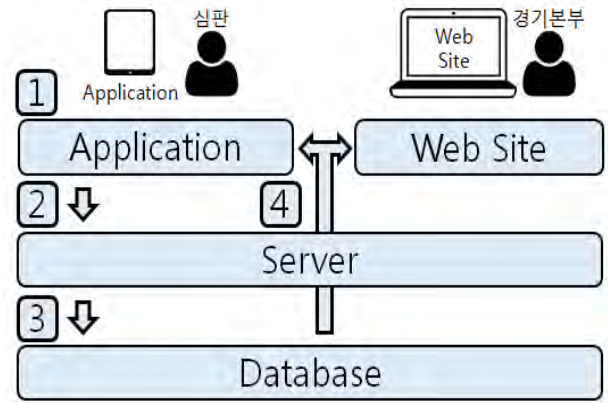
Fully Automatic Race Timing System은 경기 기록을 기록관리 프로그램으로 내보내거나 데이터베이스에 전송하는 등 효율적인 경기 기록 관리를 돕는 기능이 미비하다. 조정 경기의 기록 관리를 위해서는 경기 결과를 정확히 측정하는 것뿐 아니라 측정된 기록 정보를 효율적으로 관리하는 시스템이 필요하다.

본 논문에서는 조정경기 기록을 측정하고 측정된 기록 정보를 통합적으로 관리하기 위해 MariaDB를 이용한 조정 경기 기록 관리를 위한 애플리케이션을 구현하고자 한다.

3. 조정 경기 기록 관리 애플리케이션

본 논문에서는 MariaDB를 이용하여 조정 경기 기록 관리를 위한 애플리케이션을 구현한다. 본 논문에서 제안하는 시스템의 전체 흐름도는 (그림 1)과 같다.

(그림 1)의 (1)에서 경기장의 각 위치에서 대기하는 심판들은 각자의 위치와 용도에 따라 총 3종류의 애플리케이션 중 하나를 사용한다.



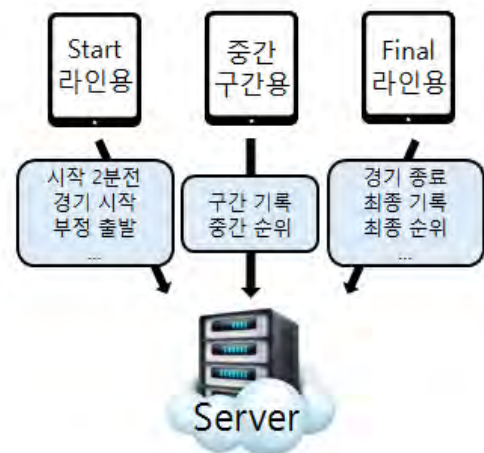
(그림 1) 전체 흐름도

애플리케이션은 스타트 라인, 중간 구간, 피니시 라인용으로 구성되어 있다. 중간 구간용 애플리케이션은 스타트 라인으로부터 각 500m, 1,000m, 1,500m 거리에 위치해 있는 심판들이 사용한다. (2)에서 서버는 각 애플리케이션으로부터 측정 기록과 경기 정보를 수신한다. (3)에서 서버는 전송받은 경기 정보와 기록을 데이터베이스에 저장한다. (4)에서 경기 운영 본부와 심판들은 데이터베이스에 저장된 경기 진행 상황과 기록 정보를 웹과 애플리케이션을 통해 확인한다.

3.1 경기 신호 제어 및 기록 측정

기존 국내에서 사용되던 기록 측정 방식에는 두 명의 심판이 필요한 반면, 본 논문이 제안하는 애플리케이션은 한 명의 심판으로도 기록 측정이 가능하다. 또한 경기 진행에 필요한 신호 제어 및 기록을 측정하고, 측정된 데이터를 서버에 자동으로 전송한다.

본 논문에서 제안하는 애플리케이션에서 각 애플리케이션에서 서버로 전송되는 정보는 (그림 2)와 같다.



(그림 2) 각 애플리케이션에서 서버로 전송되는 정보

조정 경기에서 심판은 애플리케이션을 통해 스타트 라인에서 피니시 라인에 이르기까지 500m 간격으로 구간 또는 최종 기록을 측정한다. 스타트 라인에 위치한 심판은 경기 시작 2분 전 신호, 경기 시작, 부정 출발 등의 경기 시작과 진행에 필요한 신호를 제어한다. 스타트 라인으로부터 각 500m, 1,000m, 1,500m 지점에 위치한 심판들은 해당 위치에서 구간 기록을 측정하고 중간 순위를 매긴다. 피니시 라인에 위치한 심판은 최종 기록을 측정하고 최종 순위를 매긴 뒤, 해당 경기를 종료하기 위한 경기종료 신호를 제어한다.

3.2 경기 정보 저장 및 관리

본 논문에서 제안하는 애플리케이션에서는 심판이 측정 한 기록과 경기 진행 정보를 애플리케이션을 통해 서버로 전송한다. 서버는 애플리케이션으로부터 수신한 기록 및 경기 정보를 데이터베이스에 저장한다.

서버는 스타트 라인과 피니시 라인으로부터 경기 진행과 종료에 대한 신호를 수신하고, 데이터베이스에 저장된 경기의 진행 상태 정보를 변경한다. 경기의 진행 상태에는 경기 준비, 시작, 경기 중, 종료, 부정 출발, 경기 중단 등이 있다. 또한, 서버는 500m, 1,000m, 1,500m에 해당하는 구간 기록과 순위 정보를 수신하여 데이터베이스에 저장한다. 서버는 데이터베이스에 경기 기록과 경기 정보를 저장한다.

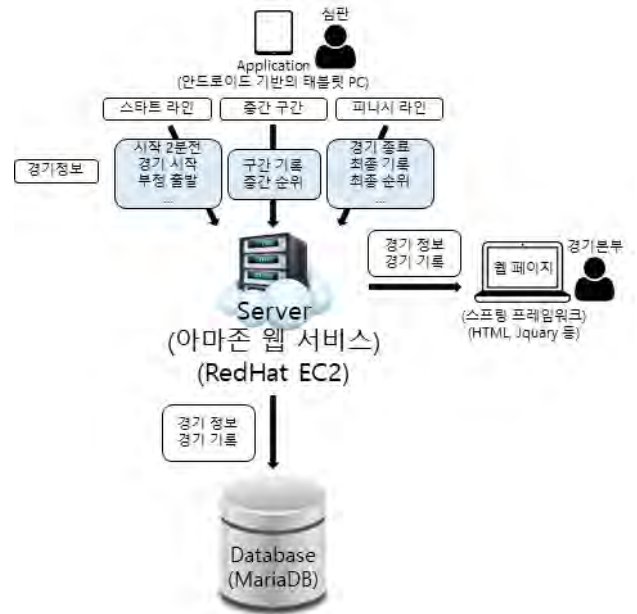
경기 운영 본부와 심판들이 경기를 원활하게 진행하고자 준비할 수 있도록 데이터베이스에 저장된 정보를 웹이나 애플리케이션으로 확인 가능하다. 각 위치에 있는 심판은 애플리케이션을 통해 경기 정보와 상황 등을 확인하고 다음 경기를 준비한다. 경기장 밖에 위치한 경기 운영 본부는 웹을 통해 경기 진행 상황과 기록 정보를 확인하고 경기를 운영한다.

4. 구현

4.1 개발환경

본 논문에서 제안한 애플리케이션은 Windows 10 운영체제에서 구현되었다. 경기장의 각 위치에서 기록을 측정하기 위해 심판들이 사용하는 애플리케이션은 Androids 기반의 태블릿PC에서 동작하도록 Java 언어로 작성되었다. 데이터 저장을 위한 서버는 아마존 웹 서비스의 RedHat EC2를 이용하고, 데이터베이스는 MariaDB를 이용해 구현하였다. 경기 운영본부에서 경기 데이터를 확인하기 위한 웹 페이지는 스프링프레임워크와 HTML, JQuery 등을 사용해 작성되었다.

본 논문에서 제안한 애플리케이션을 구현하기 위한 개발 환경은 (그림 3)와 같다.



(그림 3) 개발 환경

4.2 데이터베이스 설계

본 논문에서는 사용자가 경기 기록 측정뿐 아니라 관리자로서 경기의 정보와 기록을 관리해야 한다. 따라서 여러 정보들을 보다 효율적으로 관리할 수 있는 데이터베이스 설계가 필요하다.

본 논문에서 제안하는 MariaDB를 이용한 조정 경기 데이터 통합관리 애플리케이션의 데이터베이스 중 경기 정보와 기록을 관리하기 위한 데이터베이스 설계도는 (그림 4)와 같다.

The diagram shows two tables: 'raceinfo' and 'record'. 'raceinfo' has columns: race_num (PK, int(11)), day_race_num (int(11)), event_name (varchar(45)), race_date (varchar(45)), round_type (varchar(45)), start_time (varchar(45)), stop_time (varchar(45)), check_onoff (varchar(45)), and progression (varchar(45)). 'record' has columns: record_num (PK, int(11)), race_num (FK, int(11)), team_num (int(11)), rank (int(11)), bow_num (int(11)), 500_time (varchar(45)), 1000_time (varchar(45)), 1500_time (varchar(45)), and finish_time (varchar(45)).

(그림 4) 데이터베이스 설계도의 일부분

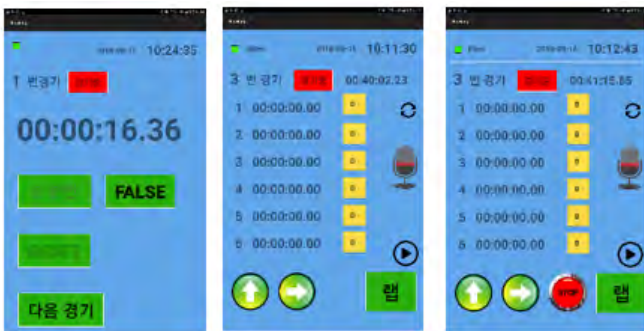
조정 경기에 대한 정보가 raceinfo 테이블에 저장되어 있으며, 측정된 경기 기록은 record 테이블에 저장되어 있다. 두 테이블은 race_num을 primary key로 하여 연결되어 있다.

raceinfo 테이블의 day_race_num은 해당 날짜의 경기 번호를 나타내고, event_name은 경기 이름, race_date는 경기 날짜를 나타낸다. round_type은 경기 타입, start_time은 경기 시작 시각, stop_time은 경기 타임을 나타낸다. start_time은 경기 시작 시각, check_onoff는 출발 준비, 경기 중, 종료 등의 경기 상황, progression은 각 경기의 진행 상황을 나타낸다.

record 테이블의 record_num은 경기 내에서의 각 조정팀 기록 번호, team_num은 조정팀 번호를 나타낸다. rank는 최종 순위, bow_num은 경기장 내에서의 lane번호, 500_time, 1000_time, 1500_time은 각각 해당 거리까지의 구간 기록을, finish_time은 최종 기록을 나타낸다.

4.3 구현 결과

본 논문에서는 경기의 신호 제어, 기록 측정, 서버로의 정보 전송 및 확인을 위한 애플리케이션과 데이터베이스에 저장된 정보를 확인하기 위한 웹 페이지를 구현하였다. (그림 5)는 제안한 애플리케이션을 위해 구현한 결과 화면이다. 각 위치에 따라 필요한 기능을 나누어 총 3종류의 애플리케이션을 구현하였다.



(그림 5) 스타트 라인, 중간 구간, 피니시 라인용 애플리케이션 화면

(그림 5)의 왼쪽은 경기 진행에 필요한 신호를 제어하기 위한 스타트 라인용 애플리케이션의 화면이다. 스타트 라인용 애플리케이션은 경기 시작 2분 전 신호, 경기 시작, 경기 중, 부정 출발, 다음 경기 등의 경기 진행에 필요한 신호를 서버에 전송한다. (그림 5)의 중간은 스타트 라인으로부터 각 500m, 1,000m, 1,500m 거리에서 사용하기 위한 중간 구간용 애플리케이션의 화면이다. 중간 구간용 애플리케이션은 해당 구간까지의 구간 기록과 순위를 측정하고 서버에 전송한다. (그림 5)의 오른쪽은 피니시 라인용 애플리케이션의 화면이다. 피니시 라인용 애플리케이션은 경기의 최종 기록과 순위를 측정하고, 경기 종료 신호를 서버에 전송한다.

호를 서버에 전송한다.

(그림 6)은 데이터베이스에 저장된 경기 정보와 기록 등을 나타내기 위한 웹 페이지 화면이다. 심판들이 애플리케이션으로 측정한 기록 및 경기 정보 등을 데이터베이스로부터 가져와 웹 페이지에 표시한다. 동시에 여러 경기가 진행되는 경우, 좌측 드롭다운 메뉴를 통해 경기별 경기 기록과 순위 결과를 확인 가능하다.



(그림 6) 경기 정보 및 기록 확인을 위한 웹 페이지 화면

5. 결론

본 논문에서는 MariaDB를 이용해 조정 경기 데이터를 통합적으로 관리할 수 있는 애플리케이션을 구현하였다. 애플리케이션을 이용해 경기 기록을 측정하고 웹과 애플리케이션으로 경기 정보와 기록을 확인 가능하다.

MariaDB를 이용한 조정 경기 데이터 통합관리 애플리케이션은 현재 타이머를 이용해 직접 측정하는 방식보다 정확한 기록 측정이 가능하며, 관리자가 직접 기록을 입력하는 과정을 자동화한다. 이를 통해 경기 정보 및 기록 관리에 드는 인력과 비용을 줄여 효율적인 기록 측정과 관리가 이루어질 것으로 기대한다.

본 논문이 제안한 애플리케이션은 현재 조정 경기에 한하여 사용이 가능하다. 그러나 후후 센서를 이용해 기록을 측정하고 카메라를 이용해 이미지프로세싱 하는 등의 기능을 추가함으로써 경마, 사이클링과 전반적인 트랙 경기와 같은 기록경기에 활용할 수 있을 것으로 기대한다.

향후 조정 경기 이외의 다른 기록경기에도 적용할 수 있도록 기능 추가 및 적용 확장 방안에 대한 연구를 진행할 예정이다.

참고문헌

[1] 대한조정협회, <http://rowing.sports.or.kr/>
 [2] Lynx System Developers, <http://www.finishlynx.com/>
 [3] FISA (The World Rowing Federation), <http://www.worldrowing.com/>
 [4] 장시웅, "소용량 데이터베이스 처리를 위한 DBMS의 성능 비교", 한국정보통신학회 논문지, Vol. 12 No. 11, pp. 1999-2004, 2008.11.
 [5] 연광제, "MariaDB DBMS 성능 향상 연구", 석사학위논문, 숭실대학교 정보과학대학원, 2017.06.