**List of Abbreviations**

AJAX – Asynchronous JavaScript and XML

API – Application Programming Interface

BPM – Beats per Minute

EMI – Electromagnetic Interference

GPS – Global Positioning Satellite

HTTP – Hypertext Transfer Protocol

IR – Infrared

NCAA – National Collegiate Athletic Association

SQL – Structured Query Language

TSDB – Time-Series Database

URL – Uniform Resource Locator

USB – Universal Serial Bus

USD – United States Dollar ($)

XML – Extensible Markup Language

1. **Approach**

The Zotikon system is composed of two subsystems - the athlete-worn device and the trainer station. The athlete-worn device collects the essential indicators about the athlete in real-time and transmits it to the trainer station for monitoring purposes. The Zotikon system assists athletes and trainers by providing real-time data monitoring on athletes while they perform. This allows for more adaptive workouts and finely-tuned recovery periods. This section details the approach used to meet the design constraints listed in Section 2. The section is composed of two major sections that divide the hardware implementation from the software implementation. Section 3.1 describes the hardware implementation for both subsystems: the athlete-worn device and the monitoring station. Section 3.2 describes the software implementation for both subsystems. A system overview is shown in Figure 3-1 that explains how the different subsystems interact to form the Zotikon system. The athlete-worn device collects the heart rate and temperature of the athlete, stores those measurements in a small internal memory until a transmission window arrives, and transmits the new measurements to the monitoring station using the mesh network inherent to Synapse. The monitoring station repeatedly polls the athlete-worn devices for new measurements and reads those measurements into the time-series database. The web server running on the monitoring station provides the single-page application to the client and an interface for the webpage to request data updates. The single-page, web application runs on a client device and requests data updates from the web server using Asynchronous JavaScript and XML (AJAX) which the application then presents in a user-friendly interface and graphs. The webpage also provides the user access to team and player data stored in the Structured Query Language (SQL). This describes the Zotikon system at a high level. The following sections provide in-depth details about each subsystem.

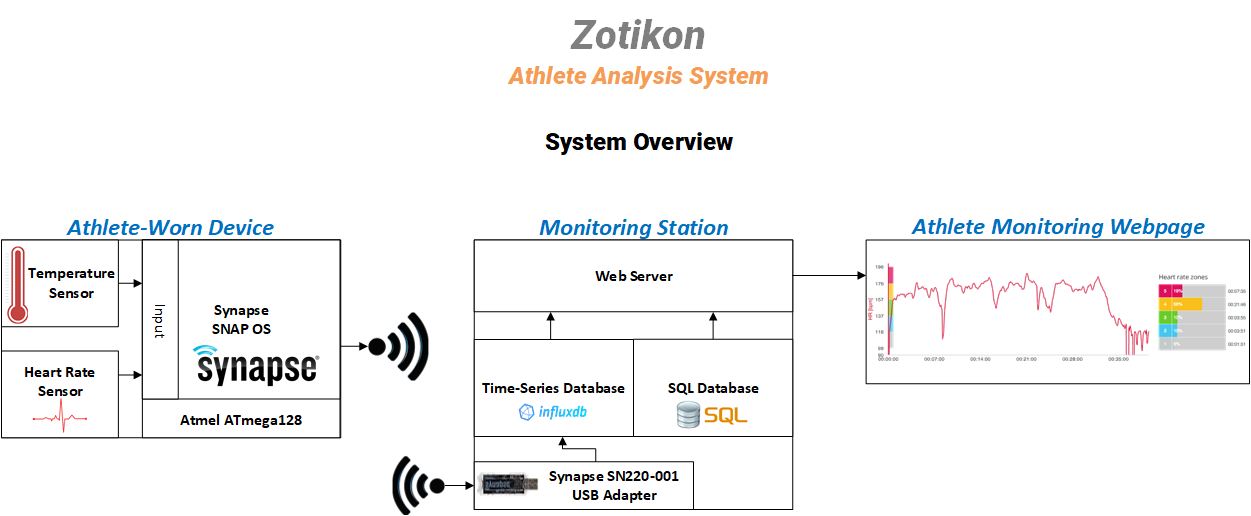


Figure 3-1 Zotikon System Overview

* 1. **Hardware**

The following subsections provide detailed explanation about each hardware component and how each component satisfies the design constraints outlined in Section 2 of this document. The hardware components were selected based on performance and physical specifications and previous team member familiarity.

* + 1. **Athlete-Worn Device**

This section details the hardware selections for the athlete-worn device.

* + - 1. **Radio**

This section describes the various radio options considered for the Zotikon system to communicate between the athlete-worn device and the monitoring station. The radio selected is the Synapse SM200. Table 3-1 shows a comparison between the key points.

Table 3-1 Radio Hardware Comparison

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Component (Radio) | Current Draw (mA) | Range (m) | Network Protocol | Cost (USD) | Bandwidth (Kbps) | Noise (dBm) | Encryption |
| Synapse SM200 | 22.5 | 457 - 762 | SNAP (mesh) | 30.07 | 250 - 2000 | -100 | AES 128-bit |
| Atmel ATmega128RFA1 | 12.5 | 457 – 762 | IEEE 802.15.4 | 6.63 | 250 – 2000 | -100 | AES 128-bit |
| Time Domain PulsON330 | 440 | 240 - 1000 | ALOHA or TDMA | --- | 19.2 – 612 | -113 to -98 | Not Implemented |
| Decawave DW1000 | 70 | 100 | Not Implemented | 15.19 | 110 - 6800 | -106 to -94 | Not Implemented |

The Synapse SM200 was chosen for the Zotikon system because the Synapse modules operate using the SNAP mesh network protocol, which provides an advantage over other products. The mesh network allows for a single Synapse module to use any other active modules to relay or boost a signal. This improves link quality and range when multiple devices are used for a team as opposed to each device making its own connection to the trainer’s station. The Synapse module with the SM200 also provides a solution to many of our design constraints. It has a built in Atmel ATmega128RFA1 microprocessor that has a 2.4 GHz radio integrated on the chip. The Atmel chip meets requirements for a microprocessor, has multiple ADCs, and has a built in python operating system. This makes the synapse module as a whole better than going with an independent chip like the Atmel or the Decawave DW1000. The Decawave is only a radio chip that is used by the Time Domain PulsON440 module. Implementing the Decawave as a standalone radio is not favorable to the Synapse module since the Synapse has the mesh networking and other hardware components. The Time Domain PulsON440 contains more hardware than needed, lacks a mesh network for wireless communication, and it consumes more power than the other options. For these reasons, the Zotikon system uses the Synapse SM200 module for its radio implementation.

* + - 1. **Microprocessor**

This section describes the various microprocessors considered for the Zotikon system to control the athlete worn devices. The microprocessor selected is the Atmel ATmega128RFA1 which is integrated in the Synapse SM200 module. Table 3-2 shows a comparison between the microcontrollers considered for the Zotikon system.

Table 3-2 Microprocessor Hardware Comparison

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Component (Microcontroller) | Cost (USD) | Current Draw (mA) | Number of GPIO | Flash Memory (KB) | Speed (MHz) | Analog to Digital Converter |
| Atmel ATmega128RFA1 | 6.63 | 14.5 | 24 | 128 | 16 | 1 ADC unit, 11 channels, 10-bit, 330 Ks/s |
| Microchip dsPIC33EP512GP806 | 7.42 | 70 | 53 | 536 | 60 | 2 ADC units, 24 channels, 10/12-bit, 1.1 Ms/s |

The Zotikon system uses the Atmel ATmega128RFA1 microprocessor as a part of the Synapse SM200 module because the Atmel chip has an integrated radio for wireless transmission, an adequate ADC, and it comes packaged in the Synapse SM200 module with a mesh networking system and built in python operating system. Using this chip with the Synapse module is easier and more efficient than trying to use a standalone microprocessor like the PIC33EP512GP806 because it requires adding a RF chip and operating system to make the whole athlete worn device operate properly.

* + - 1. **Heart Rate Sensor**

A custom heart rate detection circuit design was chosen because commercial ECG circuits were not available at a reasonable price. In order to detect the heart rate, the voltage from the heart needs to be filtered, amplified, and then sent to an ADC for processing. Instead of buying individual operational amplifiers for the amplification stage, it was more efficient to buy an instrumentation amplifier integrated circuit. This would require less space on a printed circuit board and simplify the circuit. Simply adding a resistor to the instrumentation amplifier sets the gain of the amplification stage.

Table 3-3 shows a comparison of different instrumentation amplifiers

Table 3-3 Heart Rate Sensor Instrumentation Amplifier (IA) Comparison

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Component (IA) | Voltage Rail Style | Supply Voltage (V) | Supply Current (µ) | Gain (V/V) | Gain Error  (%) | Cost (USD) |
| Texas Instruments INA126PA | Single or Dual | 2.7 - 36 | 175 | 10000 | 0.1 | 3.15 |
| Analog Devices AD623ANZ | Single or Dual | 2.7 - 12 | 375 | 1000 | 0.35 | 6.31 |
| Texas Instruments INA122P | Single or Dual | 2.2 - 36 | 60 | 10000 | 0.1 | 7.65 |

Choosing the AD623ANZ seemed clear at the time of research. It provided enough gain to amplify the range of voltages coming from the heart into a voltage that could be passed to an analog-to-digital converter [14].

Gain = Output Voltage/Input Voltage = 1.8 V/4 mV = 450 V/V Eq. 1

However, it was later found that there are better, less expensive instrumentation amplifiers that would still be able to amplify the full signal range from the heart. If more wearable devices were produced, the better choice would be the Texas Instruments INA126PA because it can perform the same functionality at a lower price.

Table 3-4 shows the comparison between components considered for analog to digital conversion.

Table 3-4 Heart Rate Sensor Analog to Digital Converter (ADC) Comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Component (ADC) | Communication Interface | Max Single-Ended Reference Voltage | Supply Voltage (V) | Resolution (bits) | Cost (USD) |
| Synapse SM200 | On-Chip | 1.8V | 3.3V | 10 | 30.07 |
| Microchip MCP3002 | SPI | 2.7 – 5.5 | 2.7 – 5.5 | 10 | 2.30 |

The Zotikon system uses the onboard ADC because it provides sufficient functionality. The ADC converts data from the heart rate, but it can handle 7 total inputs. It also provides 10 bit resolution, which provides the required accuracy for reading the desired measurements and can sample at 4 MHz, a rate that satisfies the Nyquist rate (Section 3.2.1.2 contains more details). Because the onboard ADC provides the needed functionality, paying for an external ADC is unnecessary.

* + - 1. **Temperature Sensor**

The Zotikon system uses an infrared (IR) thermometer to accurately measure the temperature of the athlete. Table 3-5 shows a comparison between components considered for temperature sensing.

Table 3-5 Temperature Sensor Comparison

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Component (Thermometer) | Price (USD) | Communication Interface | Sensor Type | Degree Accuracy (℃) | Measurement Resolution (℃) | Max VDD (V) |
| Melexis MLX90615 | 13.09 | SMBus or PWM | IR | 0.5 | 0.02 | 3.4 |
| TI TMP20AIDCKR | 1.29 | Analog Voltage | Contact | 2.5 | 0.05 | 5.5 |
| Microchip MCP9808T-E/MS | 1.19 | I2C or SMBus | Contact | 0.5 | 0.05 | 5.5 |

The Melexis MLX90615 is used in the athlete-worn devices because it offered the performance required. The IR-based sensor technology means that the temperature reading accurately reflects the athlete’s temperature, even if the sensor cannot make direct contact with the athlete. The high degree of accuracy and resolution also makes the MLX90615 a suitable choice because it allows the athlete-worn device to detect the slightest changes in body temperature. The sensor can operate on a 3.3V power supply which simplifies the design of the system by eliminating the need for various voltage rails.

* + 1. **Monitoring Station**

The monitoring station does not require a hardware approach section due to the Zotikon system being a software package that is run on any desktop or laptop computer that meets the minimum system requirements.

* 1. **Software**

The athlete-worn device collects sensor measurements and transmits those measurements to the monitoring station. The monitoring station stores the received measurements in a database that is accessed by a web server. The web server outputs dynamic web content to the viewers to monitor real-time performance on the athletes. The following subsections detail the software on each platform of the Zotikon.

* + 1. **Athlete-Worn Device**

This section details the software approach for the athlete-worn device. This device collects measurements from the sensors and transmits them to the monitoring station.

* + - 1. **Implementation**

Synapse has developed a proprietary, Python-based framework API that operates on their devices called SNAPpy. Both the athlete-worn devices and the receiver server run on SNAPpy. Zotikon makes use of this framework through their publically available SNAPpy libraries. The athlete-worn devices save immediately processed data until the monitoring station requests the data to be sent. The athlete-worn devices serialize the data according to known bit-positions and transmit the information to the monitoring station where bit-masking is used to extract the data. The API takes care of acknowledgements, message-length data, and error checking. The monitoring station polls each device at a certain interval, gathering chunks of processed data from one device at a time. If a certain polled device is out of range from the monitoring station, then the mesh network automatically resolves a path using other SNAPpy devices as intermediaries to pass the data back to the monitoring station. As the monitoring station obtains the data, it updates the time-series database with the relevant received information.

* + - 1. **Software Flow Diagrams**

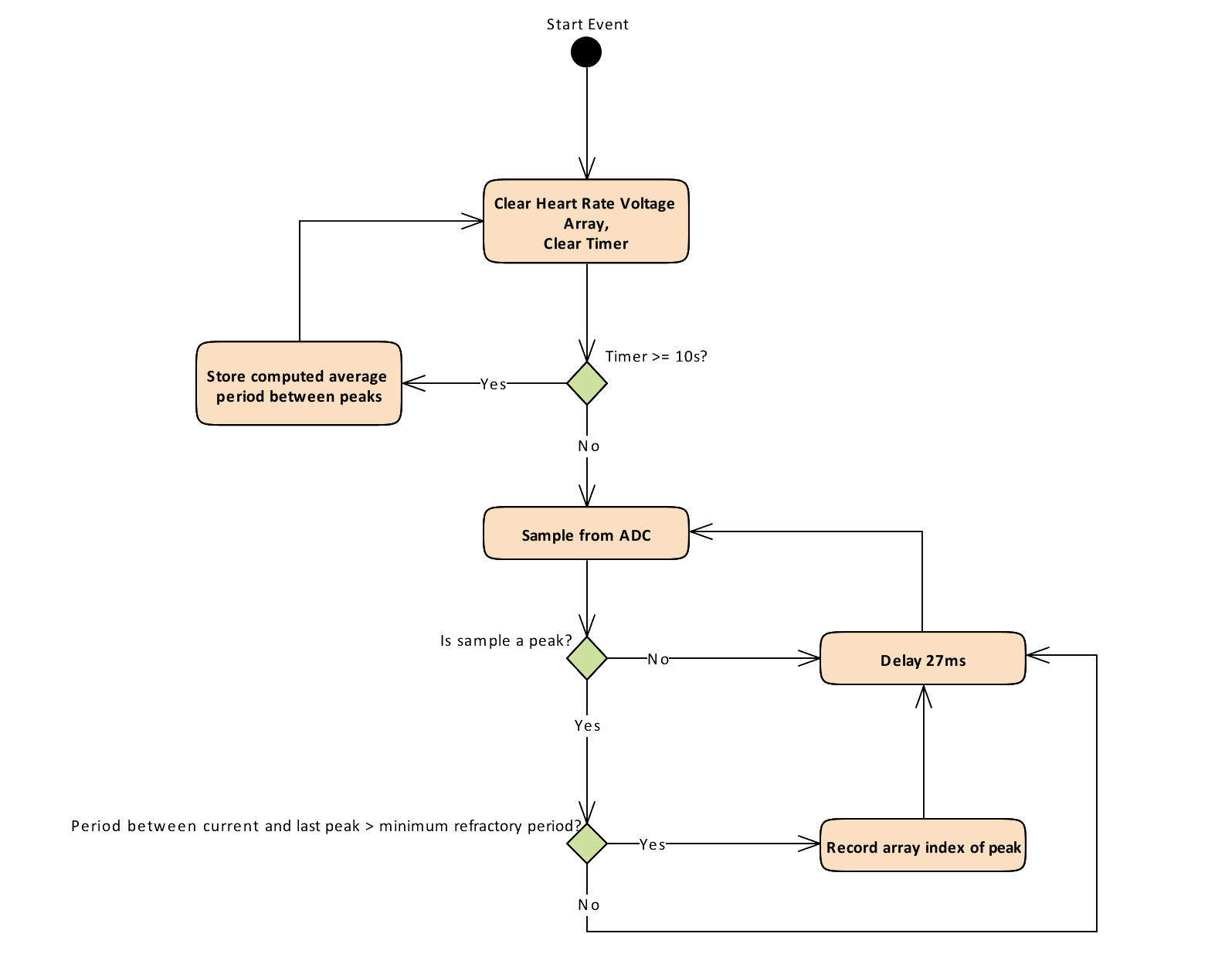


Figure 3- 2 Heart Rate Measurement Flow Diagram

Meeting the design constraint of measuring heart rates less than or equal to 220 beats per minute requires a subsystem to capture peak voltages of the heart and calculate the average period between beats. This requirement is met by measuring the voltage from the heart with the instrumentation amplifier and capturing that value with the analog-to-digital converter every 27 ms to provide a sufficient Nyquist interval (Equation 2). The algorithm in Figure 3-1 shows how the average heart rate over ten seconds is computed by detecting peak voltages that occur during a single beat of the heart. This algorithm also accounts for noise by discarding any peak values that occur after a peak and within the minimum refractory period of a heart beat [13].

fs /2 > f Eq. 2

The design constraint specifies that the device must be able to capture heart beat frequencies, f, of about 4 Hz. To ensure the constraints of the Nyquist interval are met, a sampling period of 27 ms is used. This correlates to a sampling frequency, fs, of about 37 Hz, which satisfies Equation 2.

* + - 1. **Interface Control Documentation**

The athlete-worn devices communicate with the monitoring station using the SNAPpy API. The SNAPpy interface reference [17] provides a robust framework to interact with all the devices on the Zotikon network. This interface provides all the needed functionality for the Zotikon system.

* + 1. **Monitoring Station**

This section details the software approach for the monitoring station software suite. The monitoring station is responsible for collecting the measurements from all athlete-worn devices currently in use and inserting these measurements into a database. In addition, the software suite runs a web server to serve dynamic webpages to the system viewers who are monitoring the athletes. The webpage is a single-page application that makes asynchronous calls to the web server to refresh the measurement data.

* + - 1. **Implementation**

The Synapse SN220 Universal Serial Bus (USB) adapter connects to the mesh network and uses of the SNAPpy libraries to poll the athlete-worn devices one at a time to request the latest processed data. The data is received from the devices in a serialized form with known bit-positions, and bit-masking and bit-shifting are utilized to extract the relevant data. Upon extracting data, the python script transfers the data to the time-series database to be stored.

The Zotikon monitoring station operates on two database systems, where one database manages the real-time measurement data from athletes and the other database maintains the data on players and teams that is not manipulated often. In order to effectively manage the real-time measurements gathered from the athletes, a Time-series Database (TSDB) is used to chronologically store the data. Time-series databases are designed for real-time systems that require high-speed inputs and outputs and data that is chronologically organized. Since Zotikon collects real-time data from multiple devices, a time-series database meets the needs of the system. The key features in a time-series database that are important for this system are high input speed and the ability to store data chronologically and link each data point to an individual player. A report by InfluxData [14] compares performance based on a standardized database test, as shown in Figure 3-3. Additionally, Steven Acreman published a comparison on popular TSDB implementations [15], which is show in Figure 3-4. Based on this research, InfluxDB was selected as the TSDB implementation for the Zotikon system because it meets the input-output requirements prescribed in the design constraints.

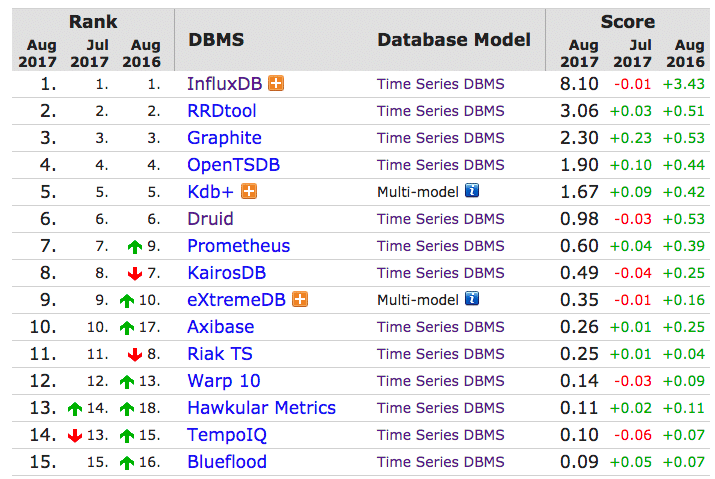


Figure 3-3 InfluxData TSDB Comparison

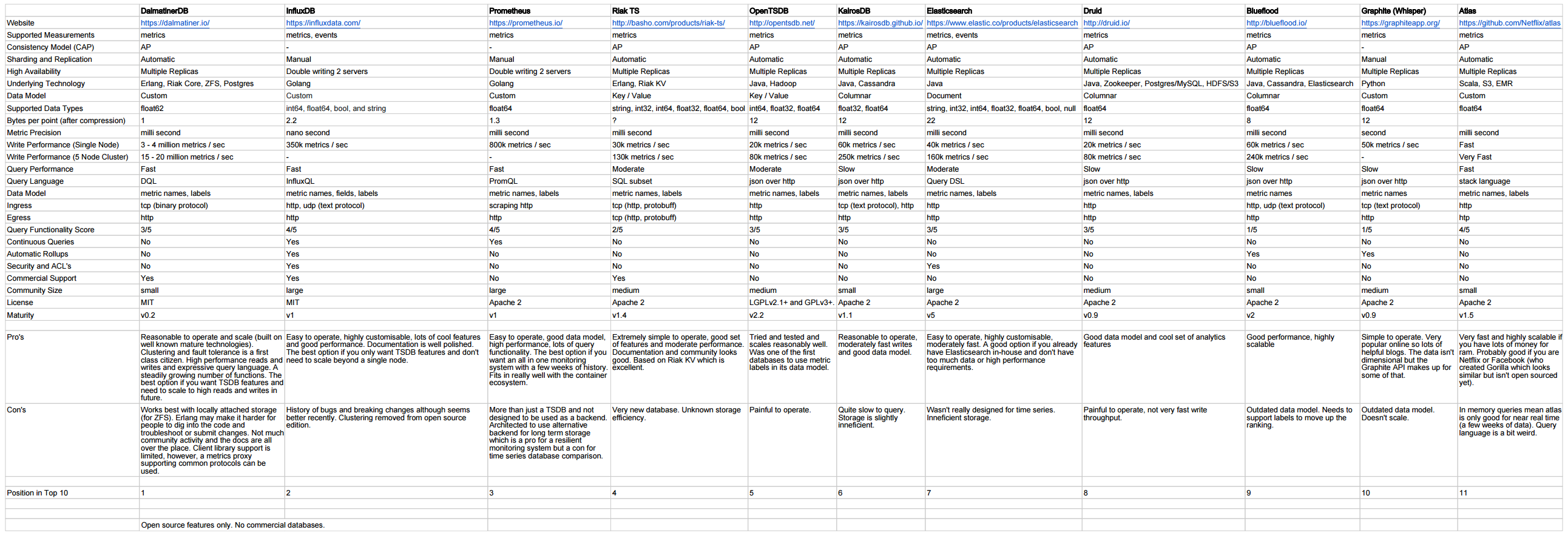


Figure 3-4 Acreman TSDB Comparison

The second database in the Zotikon monitoring station is used to manage non-measurement data that is critical to the functionality of the web application user interface. The data stored in this database is information such as user login credentials, team information, and player information. A SQL database is optimally designed to manage this data. MySQL is the implementation chosen for the Zotikon system due to programmer familiarity and no significant advantages or disadvantages compared to other SQL implementations. This database is only manipulated from the webpage application, which has permission to read and write data. The database scheme showing the relationship between SQL tables and the TSBD is presented in Figure 3-4. The SQL and TSBD are loosely-linked because there is no inherent capability to connect the databases together. The TSBD stores the unique PlayerID and EventID from the SQL tables with each data point in the EventData table. This design allows data points to be correlated to players and events for access by the web application. The SQL database structure is designed to minimize duplicate data, which determines the division of tables in the structure.

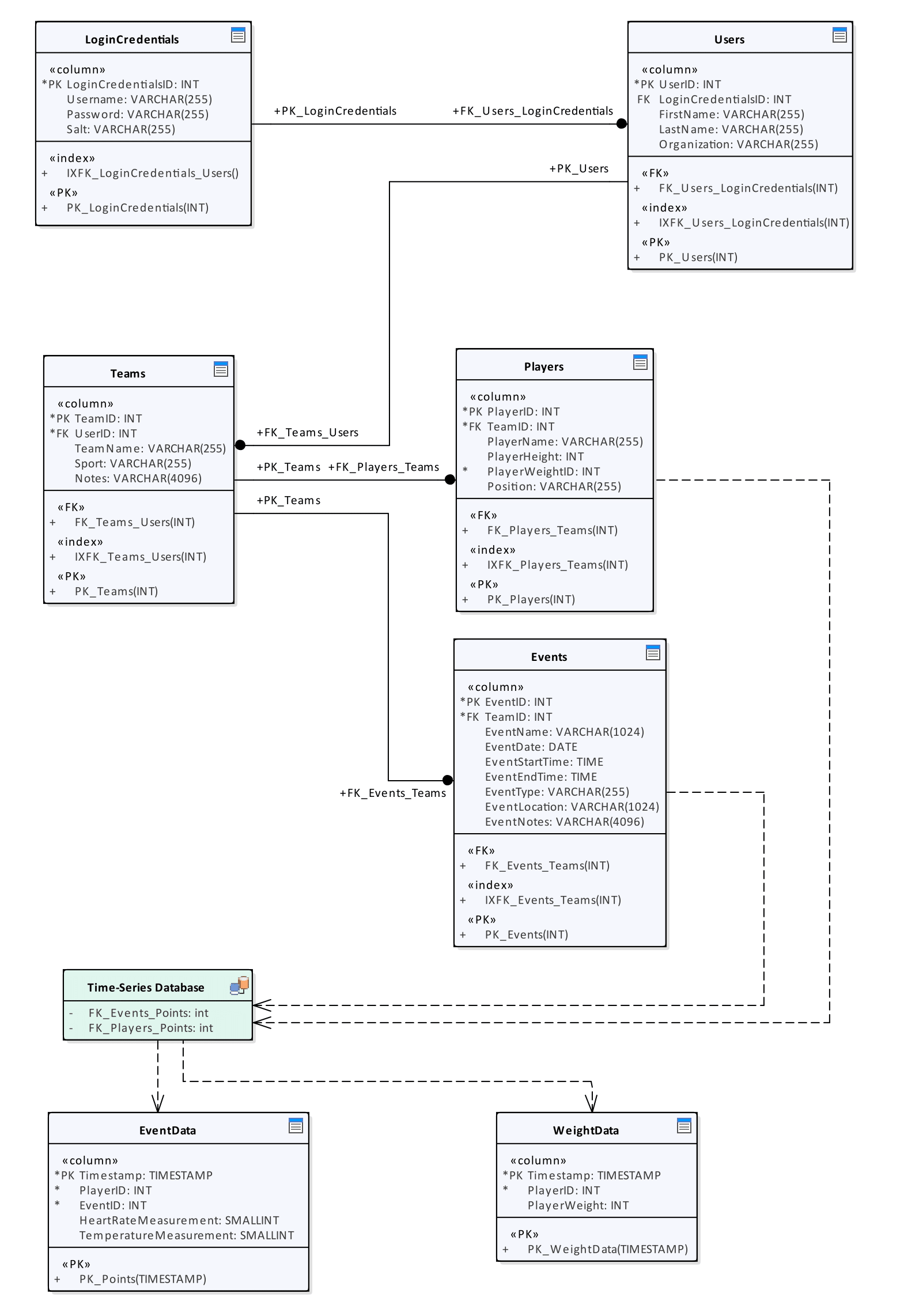


Figure 3-5 Database Design Scheme

The Zotikon system leverages a single-page web application to interface the measurements collected with the end user. The web application structure provides a well-established, cross-platform framework to present the interface and asynchronously update data elements in real time. The web application connects to the web server running on the monitoring station. The web server serves as the intermediary between the databases and the web application on the end user’s device by processing AJAX requests from the web application, fetching the requested data, and transmitting the data back to the web application. This process is shown in the software flow diagrams in Section 3.2.2.2 and the messages exchanged between the subsystems are shown in Section 3.2.2.3.

* + - 1. **Software Flow Diagrams**

Figure 3-6 shows the flow of operation for the program on the monitoring station to fetch new data from the athlete-worn devices and add that data to the time-series database.

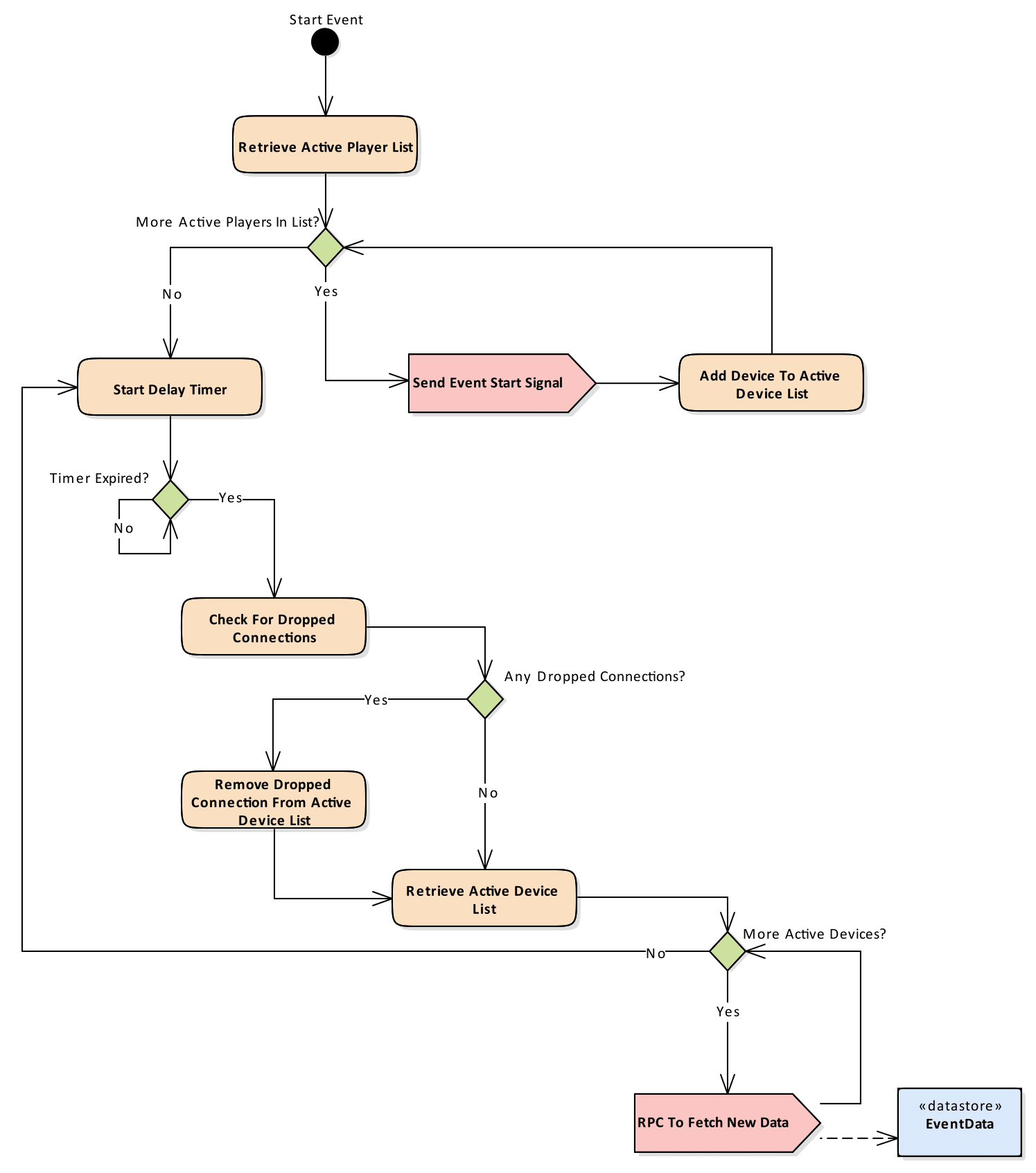


Figure 3-6 Monitor Station Data Collection Flow Diagram

* + - 1. **Interface Control Documentation**

The monitoring station runs a program to continually gather new data from the athlete-worn devices and store that data in the time-series database. As previously detailed in Section 3.2.2.1, the monitoring station interfaces with the athlete-worn devices using the SNAPpy API to collect new measurements. These measurements are written into the time-series database through the InfluxDB public Hypertext Transfer Protocol (HTTP) interface. The program uses this interface by sending a standard POST request to the query Uniform Resource Locator (URL) and appending the new data as an argument to the request. The InfluxDB online documentation for writing data to the HTTP API [16] is thorough and describes the API that is used in the Zotikon system.

The single-page web application interfaces with the web server to retrieve and update data from the databases. The web server uses the default public interfaces of the databases to request or update data and then returns the data to the webpage. To prevent invalid or malicious database manipulation, the web server processes each request and removes any damaging parameters before making the requests to the databases.

* + 1. **Use Cases**
       1. **Sunny Day**

The Zotikon system in ‘sunny day’ operation is shown in Figure 3-7. This operation occurs when all athlete-worn devices remain in contact with the monitoring station and the monitoring station remains in contact with the web application.

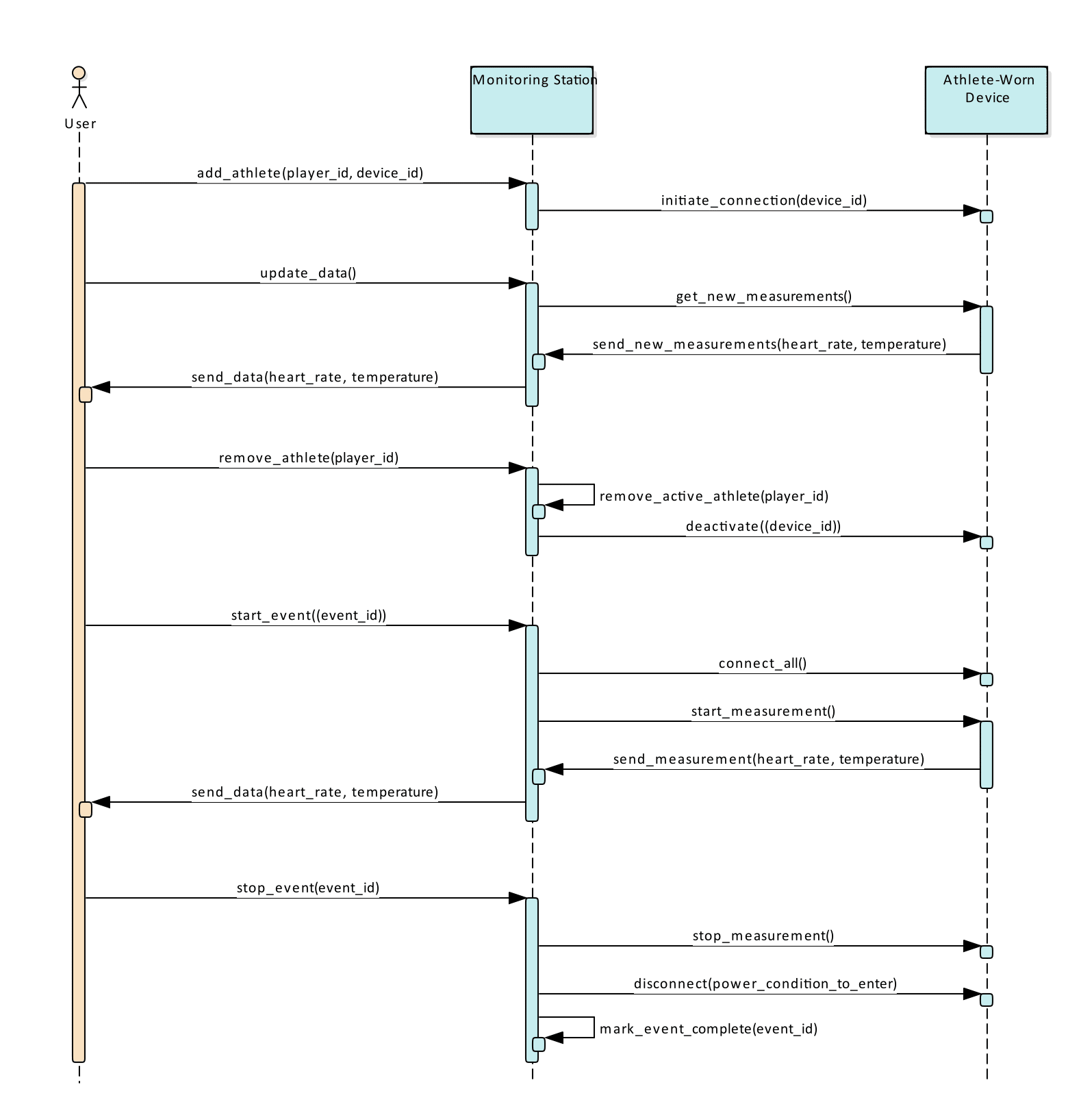


Figure 3-7 Sunny Day Use Case

* + - 1. **Rainy Day**

The Zotikon system in ‘rainy day’ operation is shown in Figure 3-8. This operation occurs when the monitoring station cannot maintain contact with all the athlete-worn devices and the system must attempt reconnection. The monitoring station retries connection at each data update interval. If connection fails, the monitoring station skips over the device and alerts the user, but maintains normal operation for all other devices.

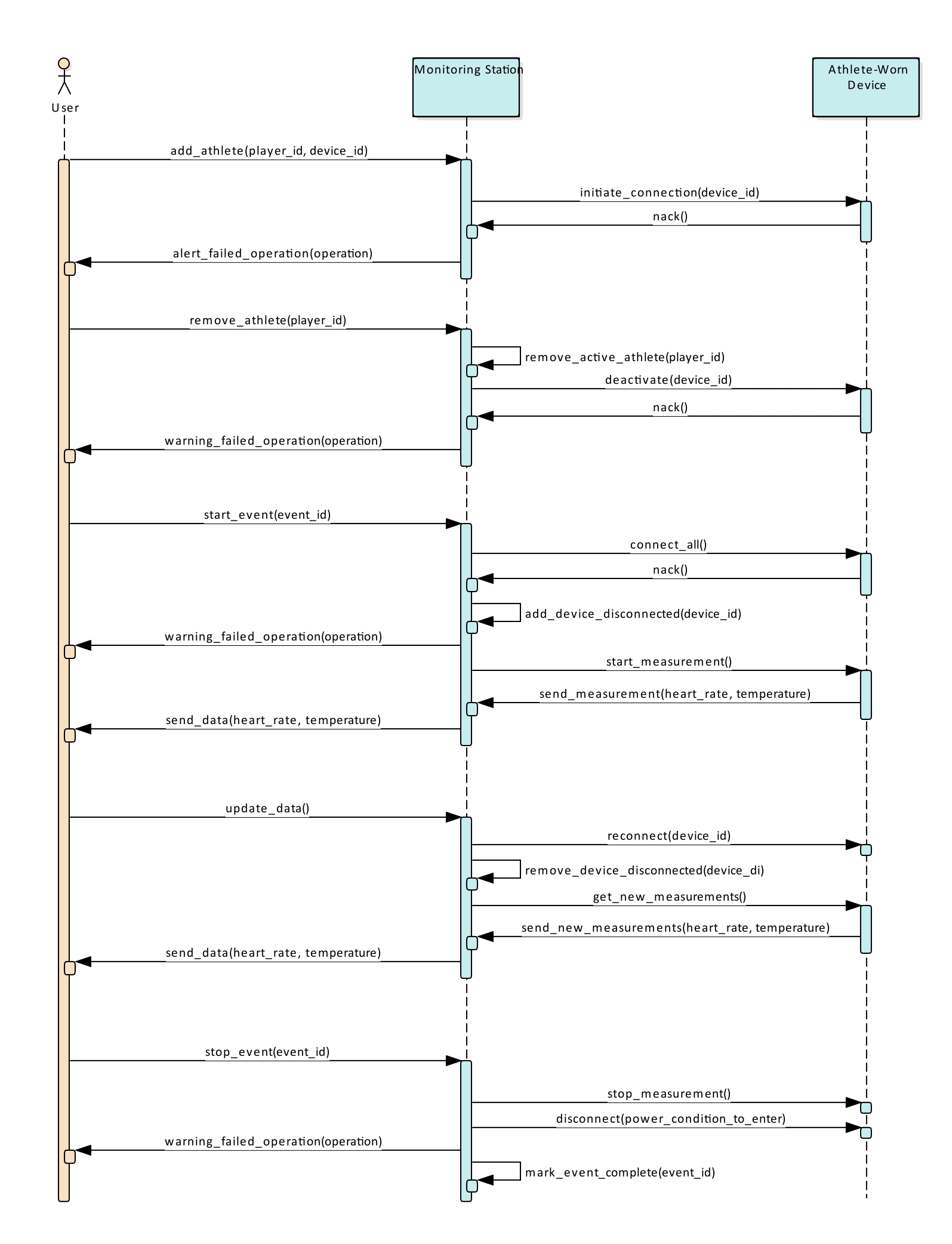


Figure 3-8 Rainy Day Use Case

References:

1. Bls.gov. (2017). Athletes and Sports Competitors : Occupational Outlook Handbook: : U.S. Bureau of Labor Statistics. [online] Available at: https://www.bls.gov/ooh/entertainment-and-sports/athletes-and-sports-competitors.htm#tab-6 [Accessed 5 Sep. 2017].
2. Team USA. (2017). Meet Team USA. [online] Available at: http://www.teamusa.org/road-to-rio-2016/team-usa/athletes [Accessed 5 Sep. 2017].
3. NCAA.org - The Official Site of the NCAA. (2017). Student-Athletes. [online] Available at: http://www.ncaa.org/student-athletes [Accessed 5 Sep. 2017].
4. Polygraphia.ca - Polygraph Examination Service. (2011). The Polygraph History. [online] Available at: http://www.polygraphia.ca/polygraph\_history.html [Accessed 31 Aug. 2017].
5. Polar.com. (2017). Innovations. [online] Available at: https://www.polar.com/us-en/about\_polar/who\_we\_are/innovations
6. ["Syracuse vs. Connecticut - Game Recap - March 12, 2009 - ESPN", ESPN.com, 2009. [Online]. Available: http://www.espn.com/ncb/recap?gameId=290710041. [Accessed: 10- Sep- 2017].
7. B. McMurphy, "Commishes agree: NCAA football games too long", ESPN.com, 2017. [Online]. Available: http://www.espn.com/college-football/story/\_/id/18421234/commissioners-college-football-coaches-seek-shorter-games-record-average-game-2016. [Accessed: 10- Sep- 2017].
8. Freitas Jr., R. (1999). Nanomedicine, Volume I: Basic Capabilities. [online] Nanomedicine.com. Available at: http://www.nanomedicine.com/NMI/8.4.1.1.htm [Accessed 11 Sep. 2017].
9. Bean, R. (2005). Facts about the skin :: Temperature of the human body. [online] Healthyheating.com. Available at: http://www.healthyheating.com/Definitions/facts\_about\_skin.htm#.WbYDZciGMkk [Accessed 11 Sep. 2017].
10. United States Soccer Federation, “United States Soccer Federation - Indoor Playing Rules”. [Online]. Available <http://www.ohsaa.org/sports/so/USSFIndoorSoccerRules.pdf> [Access 11 Sep. 2017].
11. CDC.gov, “Centers for Disease Control and Prevention.” [Online] Available at: <https://www.cdc.gov/physicalactivity/basics/measuring/heartrate.htm> [Accessed 9 Sep. 2017]
12. First In Architecture, “Average Male and Female Dimensions / Heights”. [Online]. Available: <http://www.firstinarchitecture.co.uk/average-male-and-female-dimensions/> [Accessed 11 Sep. 2017]
13. Altera.com. (2017). *Enhanced Temperature Device Support*. [online] Available at: https://www.altera.com/products/common/temperature/ind-temp.html [Accessed 11 Sep. 2017].
14. InfluxData. (2017). Time Series Database Explained | An Overview & Comparison of Top 15. [online] Available at: https://www.influxdata.com/time-series-database/ [Accessed 16 Sep. 2017].
15. Acreman, S. (2017). Top 10 Time Series Databases. [online] Outlyer Blog. Available at: https://blog.outlyer.com/top10-open-source-time-series-databases [Accessed 16 Sep. 2017].
16. Docs.influxdata.com. (2017). InfluxDB Documentation. [online] Available at: https://docs.influxdata.com/influxdb/v1.3/guides/writing\_data/ [Accessed 19 Aug. 2017].
17. Developer.synapse-wireless.com. (2017). Software Products - Developer Resources. [online] Available at: https://developer.synapse-wireless.com/software/index.html [Accessed 25 Sep. 2017].