List of Abbreviations

AJAX – Asynchronous JavaScript and XML

API – Application Programming Interface

BPM – Beats per Minute

EMI – Electromagnetic Interference

GPIO – General Purpose Input/Output

GPS – Global Positioning Satellite

HTTP – Hypertext Transfer Protocol

I^2C – Inter-Integrated Circuit

IC – Integrated Circuit

IR – Infrared

NCAA – National Collegiate Athletic Association

PCB – Printed Circuit Board

SQL – Structured Query Language

TSDB – Time-Series Database

URL – Uniform Resource Locator

USB – Universal Serial Bus

USD – United States Dollar ($)

UWB – Ultra-Wideband

XML – Extensible Markup Language

3. 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. This section is composed of two major subsections. Section 3.1 describes the hardware implementation for both subsystems, and 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 measurements to the monitoring station using the mesh network inherent to Synapse. The monitoring station continuously polls the athlete-worn devices for new measurements and stores those measurements into the time-series database. The web server running on the monitoring station provides a single-page application to the client. 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 with graphs. The web page also provides the user access to team and player data stored in the Structured Query Language (SQL). The following sections provide in-depth details about each subsystem.

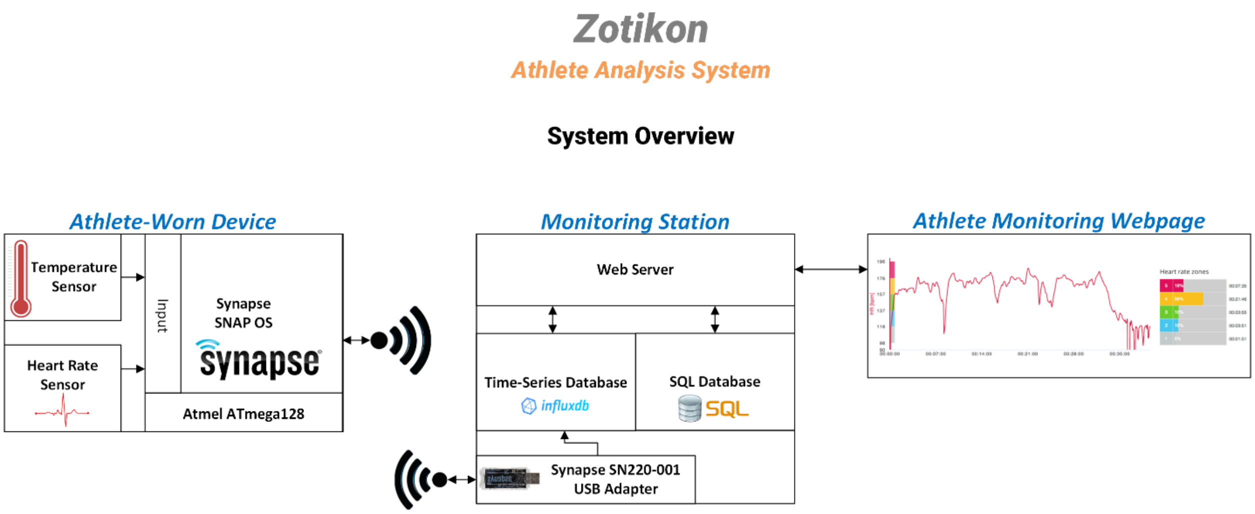

Figure 3-1 Zotikon System Overview

3.1. Hardware

The following subsections provide detailed explanations 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.

3.1.1. Athlete-Worn Device

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

3.1.1.1. 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 was the Synapse SM200. Table 3-1 shows a comparison between the radios considered for the Zotikon system.

Table 3-1 Radio Hardware Comparison

Component (Radio)	**Current Draw** (mA)	**Range** (m)	**Network Protocol**	**Cost** (USD)	**Bandwidth** (Kbps)	**Noise** (dBm)	**Transmission Frequency**
Synapse SM200	22.5	457 – 762	SNAP (mesh)	30.07	250 – 2000	-100	Narrowband: 2.4Ghz
Atmel ATmega128RFA1	12.5	457 – 762	IEEE 802.15.4	6.63	250 – 2000	-100	Narrowband: 2.4GHz
Time Domain PulsON330	440	240 – 1000	ALOHA or TDMA	1595	10 – 6800	-113	Ultra-wideband: 4.0GHz – 6.5GHz 0.5GHz Bands
Decawave DW1000	70	100	Not Implemented	15.19	110 – 6800	-106	Ultra-wideband: 3.5GHz – 6.5GHz 0.5GHz Bands

The Synapse SM200 was chosen for the Zotikon system because the Synapse modules communicate with a mesh network, consume a tolerable amount of power, provide ease of implementation, and are affordable for mass production in athlete-worn devices. The modules operate using the SNAP mesh network protocol, which provides an advantage over other products. A mesh network allows the system to transfer data using any of the athlete-worn devices in the area. Each device functions as a relay for the other devices in range to create a path from the transmitter to the receiver. This improves link quality and range when using multiple athlete-worn devices compared to each device making its own connection to the trainer's station. While the requirement to transmit 70 meters still applies to each unit to ensure connectivity if no devices are present to act as relays, the mesh topology greatly improves the transmission success rate when multiple devices are active simultaneously, such as during a team sporting event. The Synapse SM200 module also provides a solution to many other design constraints. The module operates on an Atmel ATmega128RFA1 microprocessor that has a 2.4-GHz radio integrated on the chip. The Atmel microprocessor meets the requirements for analog-to-digital converters (ADC) and Inter-Integrated Circuit (I^2C) communication channels, and it contains sufficient general purpose input/output (GPIO) pins to handle user interaction requirements. The Synapse module utilizes an embedded operating system that executes Python bytecode and supplies an application programming interface (API) to communicate with the hardware. These features make the Synapse SM200 the choice for the Zotikon system.

The Atmel ATmega128RFA1 is the microprocessor used in the Synapse SM200 module. This microcontroller was considered as a separate radio choice because it met all the design constraints for radio, microcontroller, and peripherals. However, using this device required the team to develop additional software, which the project schedule did not allow. Additionally, an embedded operating system would be required that operated on the Atmel microcontroller. The largest consideration of this device over the Synapse module was the reduced cost and lower power consumption. With all factors taken into consideration, this device was not chosen because it lacked an already implemented mesh network protocol and an embedded operating system.

The Time Domain PulsON330 is an ultra-wideband two-way ranging and communications module that offers exceptional performance in high-interference environments. This module manipulates an ultra-wideband (UWB) radio to measure distances with an accuracy of 10cm or better, which, combined with the Time Domain localization software, could have allowed the Zotikon system to track players during events. This tracking feature would have allowed for distance-travelled measurements for the athletes and would have provided the coaching staff with a unique post-game analysis method to review the movement patterns of the team. The module supports a data rate of up to 6.8Mbps, which is sufficient for the Zotikon system to transfer data measurements. The use of UWB allows the radio to be extremely resilient to interference because the transmission is spread across a 500MHz spectrum instead of being focused on a narrow channel — e.g., Bluetooth or WiFi. This would, most likely, allow for the best resilience against interference of all the devices considered. However, the module has the physical dimensions of a credit card, which is too large for an athlete-worn device, and it consumes an unfeasible amount of power for a small battery-operated device. Also, the device is too costly. While this device would have provided the Zotikon system with very unique capabilities, the size, cost, and power make this device impractical for the athlete-worn device.

The Decawave DW1000 is the UWB radio chip used in the Time Domain PulsON330 module. The chip operates using ultra-wideband frequencies with a bandwidth of 500MHz and bands ranging from 3.5GHz to 6.5GHz. Using this chip could provide the unique distance measuring features of the PulsON330 and the data rates of 6.8Mbps with a tolerable current consumption; however, since this is only a radio, additional microcontrollers would be required to implement a networking protocol to handle data transmission errors, and a program would have to be developed to perform the localization functionality. Given the design time constraints of the Zotikon system, it was not feasible to undertake the implementation of a network protocol and localization software that would be required to operate the Decawave radio as desired.

After comparing the above radios, the Synapse SM200 was selected because it sufficiently met the design constraints of the Zotikon system and simplified the design of the athlete-worn device.

3.1.1.2. 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 included in the Synapse SM200 module because the Atmel chip contains an integrated radio for wireless transmission, incorporates an adequate ADC, and comes packaged in the Synapse SM200 module with a mesh networking system and built-in Python framework. The team determined that, due to the low frequency of data collection and sufficient pin availability on the Atmel, an additional microcontroller was excessive for handling measurement collection. Using an additional microcontroller would add unnecessary complexity to the system design and consume more space on the printed circuit board (PCB).

3.1.1.3. Heart Rate Sensor

A custom heart rate detection circuit design was chosen because commercial electrocardiograph (ECG) circuits were not available at a reasonable price or size. Figure 3-2 shows the process used to measure a heart rate using contact sensors.

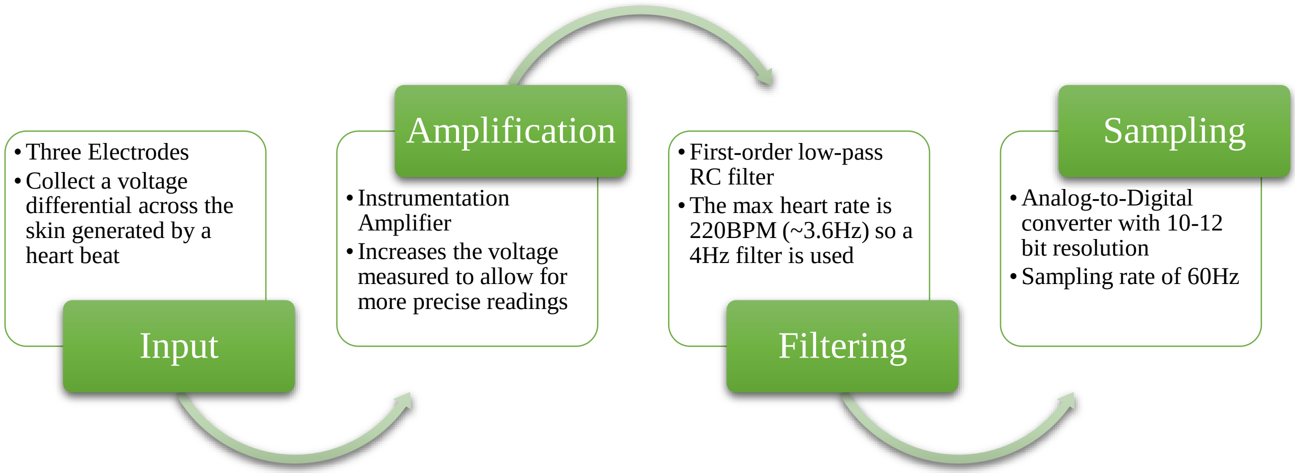

Figure 3-2 Heart Rate Measurement Process

In order to detect the heart rate, the voltage from the heart needs to be collected, amplified, filtered, and sent to an ADC for processing. The heart rate is collected using three electrodes. Two of these electrodes serve as the voltage differential inputs to the amplification stage, and the third electrode drives a constant reference voltage. The voltage differential generated by the heart can range from 1mV to 4mV. The input stage feeds into the amplification stage where an instrumentation amplifier is used to increase the voltage difference between the first two electrodes using the third electrode as a reference voltage. The difference between the two input electrodes produces the common ECG waveform showing the cycles of the heart. In order to achieve a measurable voltage from the amplifier, a gain factor of at least 450V/V must be used — this calculation is shown in Equation 1 and justified in the ADC discussion. The output of the amplification stage feeds the filter stage. This stage implements a first-order low-pass RC filter with the cutoff frequency at 4Hz. The maximum heart rate for a human is 220 beats per minute (BPM), which translates to a frequency of approximately 3.6Hz. Setting the cutoff frequency of the filter to 4Hz ensures that the heart rate signal is not attenuated, and all higher frequency noise is removed. The filtered signal is sent to an ADC to be converted to digital information usable by the microcontroller. The ADC has 10-bit resolution, which is sufficient to capture the differences in voltage. The microcontroller samples the ADC at a rate of 60Hz to ensure the Nyquist interval is sufficiently met.

The hardware for the input stage is three electrodes. The electrodes are exposed metal surfaces with wires that connect to the rest of the circuit. No special hardware is needed because the insulation-stripped end of a wire is sufficient.

The hardware needed for the amplification stage is operational amplifiers. However, instead of buying individual operational amplifiers to build the circuit, it is more efficient to buy an instrumentation amplifier integrated circuit (IC). This would require less space on a printed circuit board, provide increased performance, 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

From initial research, the Analog Devices AD623ANZ seemed to be a common choice in heart rate circuits. It provided sufficient 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]. Therefore, the AD623ANZ was chosen as the IA for the prototype development and testing. Equation 1, below, shows the calculation used to determine the desired gain factor. Based on equations from the IA datasheet, a resistor was selected to achieve the 450V/V gain factor. The maximum output voltage in Equation 1 was determined by the maximum single-ended reference voltage limitation of the analog-to-digital converter, which is described further in the next table.

$$\text{Gain} = \text{Output Voltage}/\text{Input Voltage} = 1.8\text{ V}/4\text{ mV} = 450\text{ V/V} \qquad \text{Eq. 1}$$

After receiving the Analog Devices AD623ANZ amplifiers for prototype development, the team discovered the Texas Instruments INA126PA amplifiers, which appeared to be superior and less expensive. Because the team had already begun development with the AD623ANZ amplifiers and because the project schedule did not allow time to restart the development, the team moved forward with development using the AD623ANZ. The INA126PA amplifier was noted as an improvement that could be made in a future version of the device that could possibly offer better performance at a lower cost.

The hardware needed for the filtering stage is a resistor and a capacitor. The values for these components were obtained by selecting a common capacitor value that was readily available and then calculating the resistance required to obtain the RC time constant corresponding to 4Hz. Equation 2 shows the calculation for determining resistance and capacitance for a first-order RC filter, where the cutoff frequency is 4Hz and the selected capacitor value is 10µF. Based on optimal values, the resistance needed is 4kΩ. However, the tolerances of the resistors and capacitors must be considered because they impact the frequency response of the filter. The team selected the lowest tolerances that did not significantly impact the cost of the components. The capacitor is 10% tolerant and the resistor is 1% tolerant. If the filter is designed with the above component values, then, in the worst case, the cutoff frequency becomes 3.58Hz, which will cut into the measurement region and begin attenuating signals that the system needs to measure. To prevent this, the resistor value is slightly decreased to 3.83kΩ, which is a common value and in ample supply. With this resistor value, the worst-case cutoff frequency becomes 3.74Hz, which is above the system requirement of 3.6Hz. These resistors and capacitors were available in the senior design lab.

$$\text{Frequency}_{\text{cutoff}} = (2\pi RC)^{-1} \qquad \text{Eq. 2}$$

The hardware needed for the conversion stage is an analog-to-digital converter. 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** (V)	**Supply Voltage** (V)	**Resolution** (bits)	**Cost** (USD)
Synapse SM200	On-Chip	1.8	3.3	10	30.07
Microchip MCP3002	SPI	2.7 – 5.5	2.7 – 5.5	10	2.30

The Zotikon system uses the analog-to-digital converter onboard the Atmel ATmega128RFA1 because it provides sufficient functionality to meet the design constraints. While the system only requires one ADC input, it can handle 7 total inputs. It also provides 10-bit resolution, which is the required accuracy for reading the heart rate measurements, and it 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 and consuming more PCB space is unnecessary.

3.1.1.4. Temperature Sensor

The Zotikon system uses an infrared (IR) thermometer to accurately measure the core 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** best (worst) (°C)	**Measurement Resolution** (bits)	**Max V_{DD}** (V)
Melexis MLX90615	13.09	SMBus or PWM	IR	0.5 (3.0)	15	3.4
TI TMP20AIDCKR	1.29	Analog Voltage	Contact	0.5 (2.5)	12	5.5
Microchip MCP9808T-E/MS	1.19	I2C or SMBus	Contact	0.5 (1.0)	10	5.5
Silicon Labs SI7051-A20-IM	1.95	I2C	Contact	0.1 (0.25)	16	3.6

The Melexis MLX90615 was used in the prototype athlete-worn device because it offered the performance required to conform to most of the design constraints. Per these constraints, the measurable temperature range must include 15°C to 40°C and the measurement accuracy must be 0.25°C. The MLX90615 uses IR-based sensor technology, which means that the temperature reading accurately reflects the athlete's temperature even if the sensor is not in direct contact with the athlete. Due to the IR sensor, the MLX90615 does not perform with the required accuracy. This fact is mitigated because it is rated for use in medical thermometers, so it was sufficient to measure the temperature of athletes. The slightly worse accuracy is acceptable because the IR technology maintains an accurate temperature reading even if the device gets separated from the athlete. The sensor can operate on a 3.3-V power supply, which simplifies the design of the system by eliminating the need for multiple voltage rails.

The Silicon Labs SI7051-A20-IM is a contact sensor that meets the temperature accuracy and range constraints with an improvement in resolution. While the accuracy is an advantage over the MLX90615, the contact sensor poses a serious design problem because the surface of the device must be exposed to the athlete. This presents challenges with the IP64 constraint because the electronic device is exposed to natural elements, such as moisture.

3.1.1.5. Power Requirements

The athlete-worn device is a battery-powered subsystem that must meet a runtime constraint of four hours. In order to ensure the device operates for the required runtime, the team performed a power analysis of the components in the athlete-worn device. The components that consume power in the device are the radio, instrumentation amplifier, and temperature sensor. Each of these components has a rate maximum current consumption. Based on this current rating, a power consumption rating was generated by multiplying the current consumption times the runtime requirement. The power calculations are shown in Table 3-6. By summing the power consumption of the components, the athlete-worn device total power consumption is 97.92mAh. This determines the battery capacity needed to power the device to power for the minimum runtime. The two important considerations for the battery selection process are the capacity and the form factor. The capacity is necessary to provide enough power to run the device for the minimum runtime. The form factor is necessary because the battery must meet the physical size constraints for the device.

Table 3-6 shows a comparison between the batteries considered for the device.

Table 3-6 Power Consumption Calculations

Radio	max current = 22.5mA ➔ max power = (22.5mA * 4 hours) = 90mAh
IA	max current = 0.480mA ➔ max power = (0.480mA * 4 hours) = 1.92mAh
Temperature Sensor	max current = 1.5mA ➔ max power = (1.5mA * 4 hours) = 6mAh
Athlete-Worn Device	max power = (90mAh + 1.92mAh + 6mAh) = 97.92mAh

Table 3-7 Battery Comparison

Component (Battery)	**Price** (USD)	**Capacity** (mAh)	**Output Voltage** (V)	**Form Factor**	**Chemical Composition**
Panasonic CR3032	2.45	500	3.0	Coin Cell	Lithium
Duracell 2430	4.00	285	3.0	Coin Cell	Lithium
Energizer CR2430	3.40	290	3.0	Coin Cell	Lithium
Duracell AAA	1.00	1150	1.15	Coppertop	Alkaline

The Panasonic CR3032 battery was selected for the athlete-worn device. The capacity of this battery was five times the required amount, but the price of the battery was competitive with other smaller batteries on the market. The output voltage of 3.0V is sufficient to drive a voltage regulator to power the subsystem.

3.1.2. Monitoring Station

The Zotikon monitoring station does not require any hardware development because it is a software package that is run on any desktop or laptop computer that meets the minimum system requirements.

3.2. Software

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

3.2.1. 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.

3.2.1.1. Implementation

Synapse has developed a proprietary, Python-based framework API that operates on their devices called SNAPpy. Both the athlete-worn devices and the monitoring station utilize SNAPpy to communicate. Zotikon makes use of this framework through the publically available SNAPpy libraries. The athlete-worn devices save immediately processed data until the monitoring station requests the data to be sent. The 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.

3.2.1.2. Software Flow Diagrams

The following figure describes the measurement collection process on the athlete-worn device.

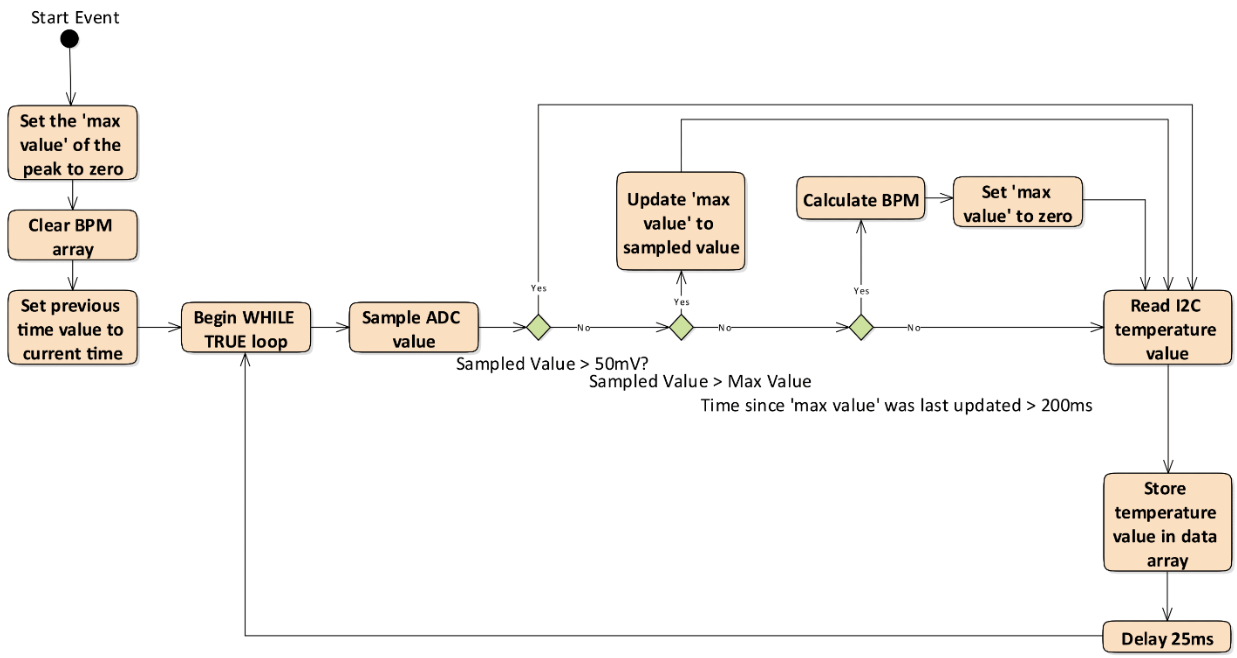

Figure 3-3 Data Measurement Flow Diagram

The athlete-worn device enters the data collection state immediately after receiving an ‘event start’ condition from the monitoring station. The data collection state has two objectives — measure the heart rate and read the temperature from the athlete. The heart rate measurement is accomplished using a peak detection algorithm to find the time between peaks in the athlete’s pulse. The heart has a minimum refractory period of approximately 250ms in healthy humans; so, the peak detection algorithm utilizes this fact. The algorithm begins by finding the first observed peak in the pulse. If the device does not observe a higher peak within 200ms, then it concludes that one pulse has occurred. The time of the current pulse’s peak is compared with the time from the previous peak to determine the pulse rate, which is stored in the athlete-worn device data buffer. This process is repeated on a regular interval until an ‘event stop’ condition occurs. The algorithm provides a few simple noise protections to prevent unwanted peaks.

The design constraint specifies that the device must be able to capture heart rate frequencies, f, up to 220BPM, which correlates to approximately 4Hz. To ensure the constraints of the Nyquist interval are met and to follow common engineering practices, a sampling period of 10 times the requirement was selected. This equates to a sampling frequency, fs, of about 40Hz, which satisfies Equation 3. To meet this sampling frequency, the microcontroller samples the ADC every 25ms.

$$f_s / 2 > f \qquad \text{Eq. 3}$$

The temperature is collected from the thermometer at the same frequency as the heart rate. This choice simplifies the data collection process and provides sufficient frequency for the temperature measurements to accurately represent the temperature changes an athlete may experience.

3.2.1.3. Interface Control Documentation

The athlete-worn devices communicate with the monitoring station using the SNAPpy API. The SNAPpy interface reference [16] provides a robust framework to interact with all the devices on the Zotikon network. This interface provides all the needed communication functionality for the Zotikon system.

3.2.2. 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 in use and inserting these measurements into a database. In addition, the software suite runs a web server to serve dynamic web pages to the system viewers who are monitoring the athletes. The web page is a single-page application that makes asynchronous calls to the web server to refresh the measurement data.

3.2.2.1. Implementation

The Synapse SN220 Universal Serial Bus (USB) adapter connects the monitoring station to the mesh network and uses the SNAPpy libraries to poll the athlete-worn devices to request the latest processed data. The data is received from the devices in a serialized form with known bit-positions, which allows bit-masking and bit-shifting to extract the relevant data. Upon extracting the data, a 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 the athletes and the other database maintains the data on players and teams that is crucial for the functionality of the user interface. 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. A TSDB far exceeds the necessary constraints of the Zotikon system, but it was selected to promote the scalability of the system and prevent future design changes in the event that more devices are used simultaneously. The key features in a TSDB that are important for this system are high input speed as well as the ability to store data chronologically and link each data point to an individual player and event. A report by InfluxData [14] compares database performance based on a standardized TSDB test, and the results stated that InfluxDB was the best time-series database. Additionally, Steven Acreman published a comparison on popular TSDB implementations [15], which included a thorough data analysis that presented all the information to the reader, and this test placed InfluxDB as the second best time-series database. The first-place TSDB in Acreman's report was DalmatinerDB because of its impressive write performance, but it lacked a large online community and had a slower query speed score. Based on this research, InfluxDB was selected as the TSDB implementation for the Zotikon system because it meets the input and output requirements described in the design constraints. The InfluxDB API is user-friendly and provides the necessary database access. Additionally, the InfluxDB online documentation and forums are well-written with essential information about database operations.

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 web page application, which has permission to read and write data. The database scheme showing the relationship between SQL tables and the TSDB is presented in Figure 3-4. The SQL and TSDB are loosely linked because there is no inherent capability to connect the databases. The TSDB 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.

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. It processes AJAX requests from the web application, fetches the requested data, and transmits the data back to the web application. The messages exchanged between the subsystems are discussed in Section 3.2.2.3.

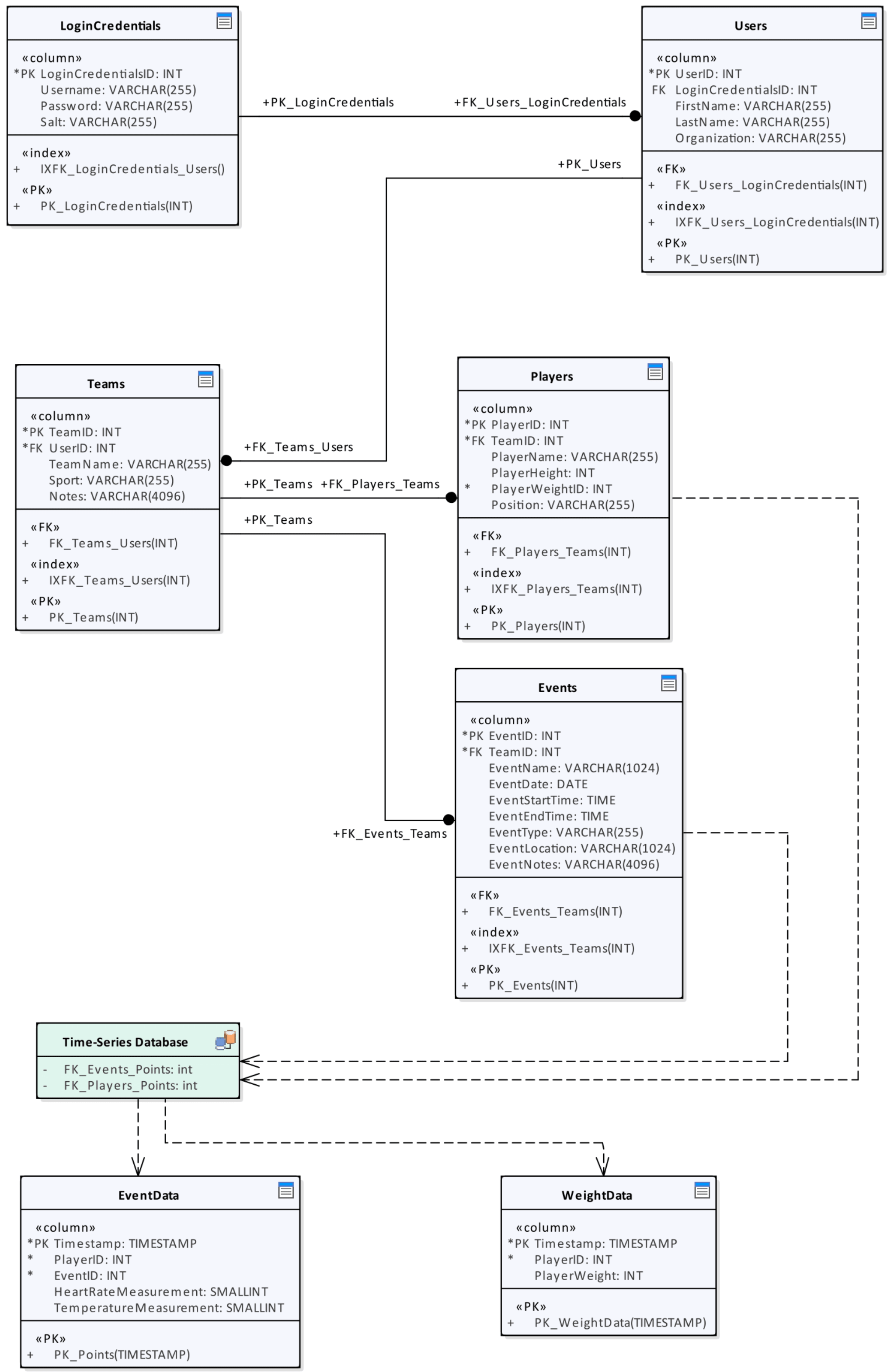

Figure 3-4 Database Design Scheme

3.2.2.2. Software Flow Diagrams

Figure 3-5 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.

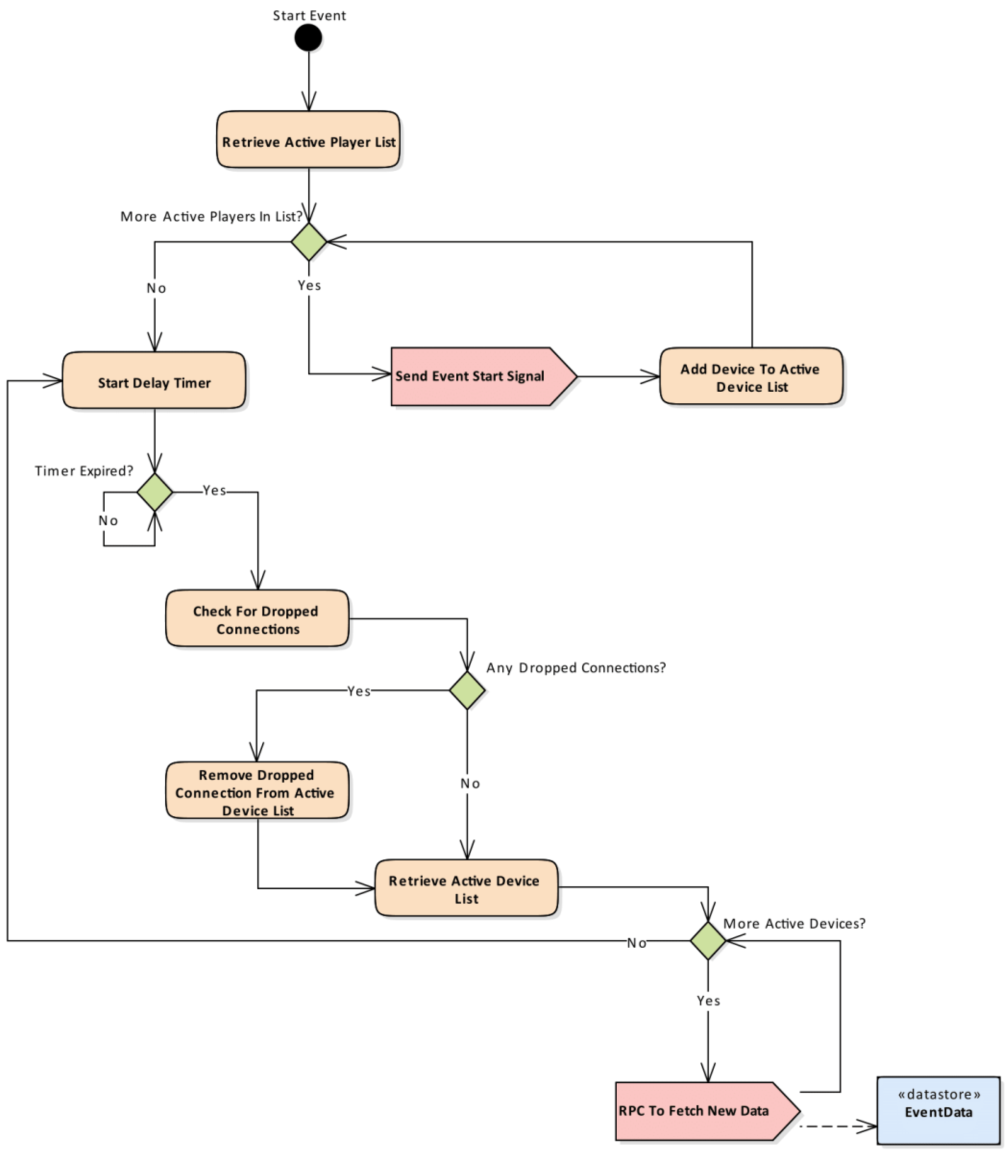

Figure 3-5 Monitor Station Data Collection Flow Diagram

3.2.2.3. 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 [17] 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 web page. 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.

3.2.3. Use Cases

This section details the sunny and rainy day use cases that may occur during the operation of the Zotikon system.

3.2.3.1. Sunny Day

The Zotikon system in 'sunny day' operation is shown in Figure 3-6. 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.

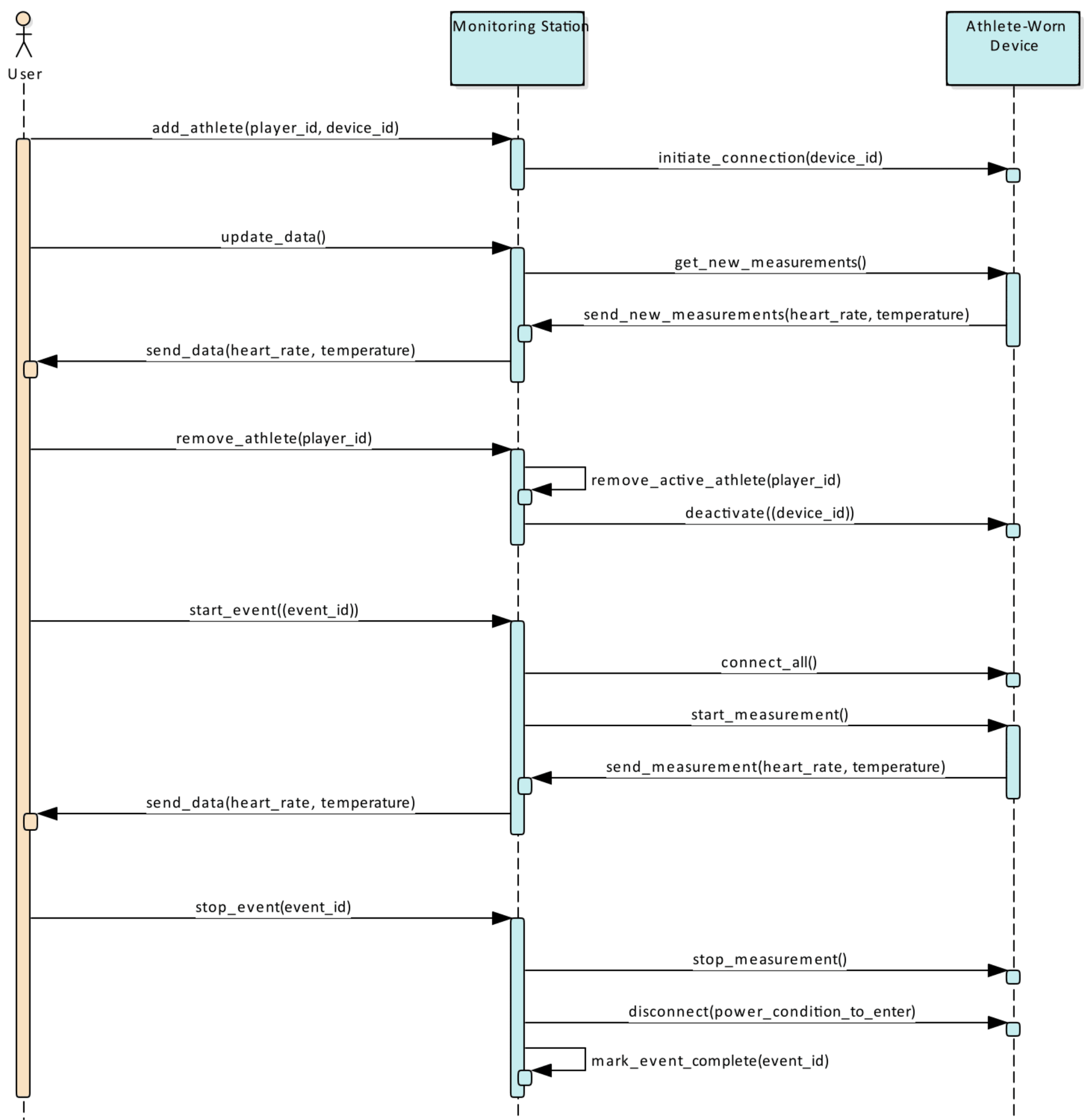

Figure 3-6 Sunny Day Use Case

3.2.3.2. Rainy Day

The Zotikon system in 'rainy day' operation is shown in Figure 3-7. 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 the 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.

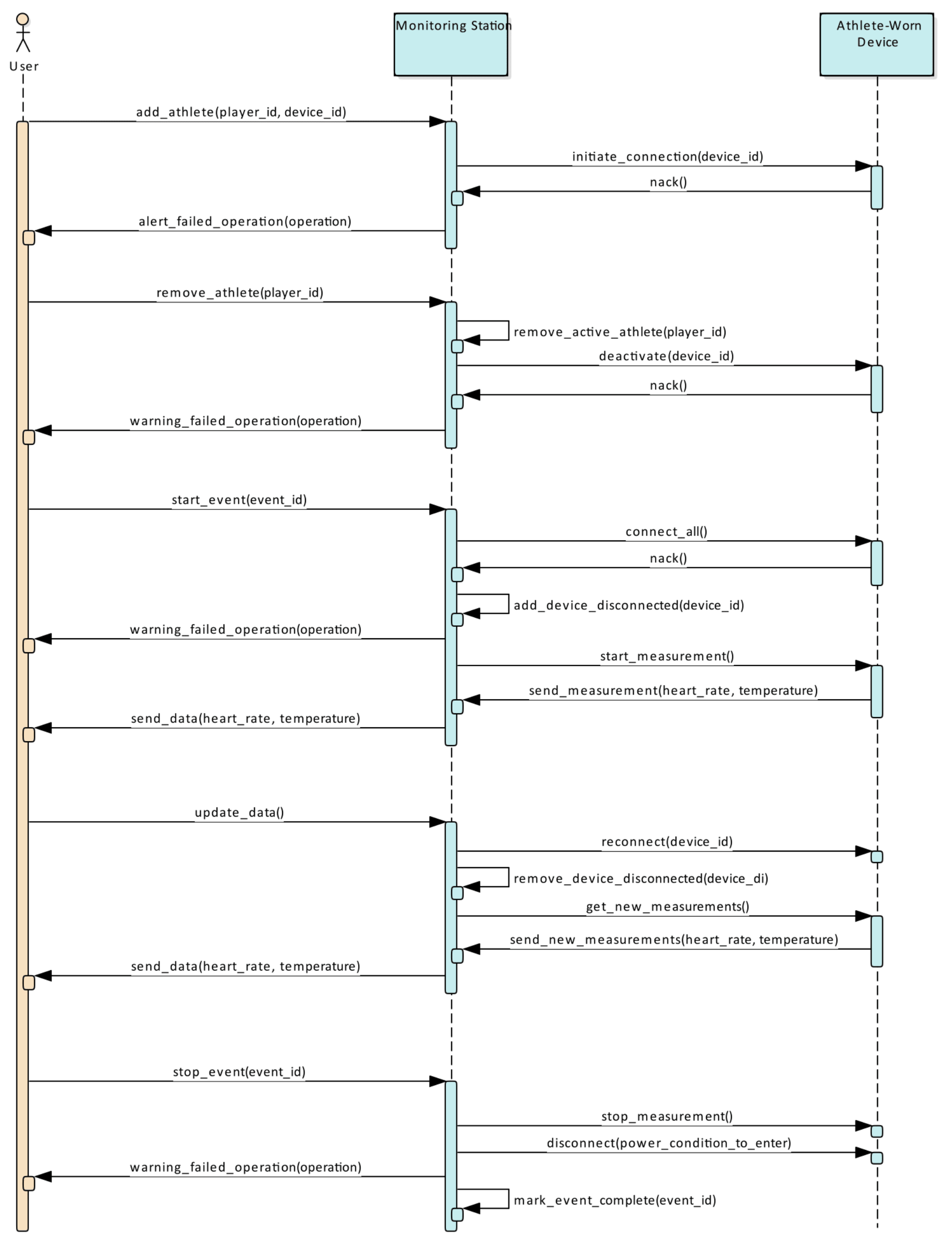

Figure 3-7 Rainy Day Use Case

References:

[1] NCAA.org - The Official Site of the NCAA. (2017). Student-Athletes. [online] Available at: http://www.ncaa.org/student-athletes [Accessed 5 Sep. 2017].

[2] 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].

[3] Team USA. (2017). Meet Team USA. [online] Available at: http://www.teamusa.org/road-to-rio-2016/team-usa/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] Developer.synapse-wireless.com. (2017). Software Products - Developer Resources. [online] Available at: https://developer.synapse-wireless.com/software/index.html [Accessed 25 Sep. 2017].

[17] Docs.influxdata.com. (2017). InfluxDB Documentation. [online] Available at: https://docs.influxdata.com/influxdb/v1.3/guides/writing_data/ [Accessed 19 Aug. 2017].