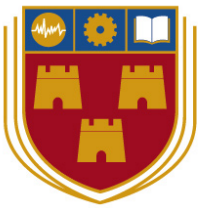


INSTITUTE of TECHNOLOGY CARLOW

FOALARM

FOALING ALARM

RESEARCH DOCUMENT

DAVID KELLY | C00193216

Bachelor of Science (Honours) Software Development

Supervised by Nigel Whyte. Date: 7th November 2017.

TABLE OF CONTENTS

ABSTRACT

The purpose of this document is to detail the research carried out as part of my 4th year Software Development Project module. The following sections are an in depth analysis of the current market needs; existing products within the foal predicting technology space; and, the programming languages, frameworks and tools considered to build my solution. As a result of this document I will be able to confidently select the technologies best suited to implement my solution.

INTRODUCTION

This paper is an in depth account of the research carried out in support of my 4th year software development project. I will begin by discussing the current state of the industry; requirements of horse breeders and mare owners; and, the justification for this project's development. It is important to understand current practices and trends within the equine breeding industry to better understand the challenges faced by potential users so to develop a solution that meets their expectations.

The aim of this project is to create a discrete and universal equine wearable foaling alarm with real-time notifications to remote users via an app.

In order to select suitable technological solutions for the build, functionality of existing products was to be investigated. The success of this project will be in its ability to address the limitations of exiting products while also matching their core advantages. I will discuss the most significant products currently available, their advantages, limitations and my own user experience. I will also discuss foaling predictors to fully consider all methods and measures of predicting imminent foaling. This will provide me with a foundation to gauge this project's scope.

Subsequently, I will consider the components, technologies, platforms and tools which I can potentially use including:

- Hardware
- Databases
- Web & App Development
- Mobile App Development
- Programming Languages
- Development Tools
- Testing
- Documentation
- Deployment

As a result, I intend to conclude the most suitable path to this project's development.

INDUSTRY NEEDS

Horse breeding is a high-risk, high reward business. Despite the fact that horses have the second highest mortality rate in the animal kingdom [CK09] the equine industry is failing to develop technological advancements in the area of mare monitoring and foaling prediction. Attended foal deliveries are essential to ensuring the safety of both mare and foal. Traditionally, breeders anticipate sleepless nights in the days or weeks prior to delivery as 80% of mares will foal between 11 pm and 3 am. Difficult births, called dytocias, account for 10% of all foal births [PW16]. This incidence rate drastically varies by breed with thorough-bred, Arabian, draught and miniature horses being the most at risk. One breed in particular; American miniature mares require assistance in 70% of births.

According to Pam Wilkins, DVM, most dytocias are easily corrected by good foaling managers or veterinarians on the farm. It is the ones that go one for more than 20 minutes where most fatalities occur. The chance of losing a foal increase 16% for each 10-minute delay in getting a foal out after 20 – 30 minutes.

"Early detection and rapid appropriate intervention are critical to foal survival in an equine dystocia." [PR12]

As horse breeds advance through selective breeding the risks increase for mares, unborn foals and financial stakeholders. Technology has enabled breeders to remotely monitor mares and today, visual monitoring is the primary method of predicting foaling and ensuring an attended delivery. The majority of breeders rely on closed-circuit or IP cameras to allow them watch pregnant mares in the weeks close to parturition. This level of dedication is not sustainable for the average person who may have commitments outside of the equine industry. Professional foaling centers typically reply on 24-hour staff to watch pregnant mares and despite the cost, mare owners are willing to pay in order to guarantee an attended delivery and protect themselves from loss.

The use of sensors has enabled breeders to monitor mares and continue their daily lives (within reason). Foaling alarms that utilize position sensors, thermometers or magnets have reduced the need for 24-hour visual monitoring by notifying breeders of possible foaling events. Each of these solutions attempt to address the need for constant monitoring but none do so without significant limitations. Limitations which will be discussed under 'Existing Products'.

EXISTING PRODUCTS

Equipage or Breeder-Alert

The Equipage by Keyport and the Breeder-Alert manufacture identical technology. Both devices consist of a transmitter, repeater and pager. The transmitter is enclosed in a pouch which attaches to the underside of the mare's head on her halter. It identifies foaling events when the mare is in lateral recumbency for 10 seconds. A signal is emitted to the nearby repeater, which activates the pager. The mare and the pager must be within 500 feet or 2-miles of the repeater station respectively. Both systems do have the option of adding an automatic telephone dialler to the repeater unit which transmits a pre-recorded message to up to four telephone numbers. The dialler plugs into the back of the repeater and is activated when the repeater receives a signal form the transmitter.

Multiple mares can be monitored at one time with additional transmitters. The repeater and pager can receive signals from up to 20 transmitters.

Cost

Complete Equipage system to monitor 1 mare: €504.22

Additional transmitter: €75.41

Advantages

- The transmitter is non-invasive; it does not require a veterinarian or special training to attach to the mare.
- Installation and set-up of both systems are relatively simple.
- The transmitter is very low profile.
- Scalable; multiple mares can be monitored at one time (max. 20).

Disadvantages

- Both systems require the users to be on-site (remaining within a two-mile radius of the repeater station).
- The system relies on radio frequency to signal the repeater and activate the pager. Due to the construction of horse buildings being predominantly steel or concrete radio frequency signals to activate the pager can be negatively affected.
- The systems do not distinguish individual mares. Although up to 20 mares may be monitored at one time and alarm signal will require the user to check each mare to confirm the source of the alarm.
- Some mares may not lie down to foal.
- Late term mares may lie down to sleep, this will result in repeated false alarms.

User Remarks

I have used the Equipage system since 2014 to monitor mares. The mares (transmitters) and the repeater are located in the same barn. The barn is situated approximately 250ft from the residence where the pager would predominately be based. Each mare has a CCTV camera overhead which allows us to monitor her from the house.

Typically, we will monitor up to three mares concurrently with the Equipage system, so in the event of the pager being activated we will need to check the three mares to determine the source of the alarm. This process simply involves switching on a nearby television to view the camera feed. A small number of mares is relatively easy as the television will typically split the screen to accommodate multiple feeds. Usability of the Equipage is therefore heavily dependent on the users having access to visual monitoring. Imagine monitoring multiple mares without the use of cameras, where every pager alarm would result in

the user physically checking each mare to determine the source. When you multiply this by potentially numerous false alarms per night you question the systems advantages over traditional visual monitoring.

Although the repeater is located within the two-mile radius limit, communication between the repeater and pager is unreliable. I believe that the metal construction of the barn as well as trees between the house and barn are greatly reducing the transmitting distance of the repeater. This has resulted in the system not activating the pager when a mare goes into lateral recumbency and as a consequence, on one occasion, a fatal delivery.

Another significant drawback of the Equipage and Breeder Alert is the lack of remote pager signally as the pager needs to remain in close proximity to the repeater and transmitters. Neither systems allow for users to continue their daily routine whilst monitoring expectant mares.

I do believe that the Equipage and Breeder Alert systems are one of surest means of ensuring attended deliveries but they do not come without significant limitations.

Image courtesy of Kee-Port Inc. [EP14]

Foalert or Foaling-Alarm

The Foal-Alert or Foaling-Alarm both manufacture a system comprised on an equine wearable, repeater and pager. The wearable is a small device that is sewed on to the mare's vulva. It is comprised of a magnet and transmitter. The physical separation of the vulva lips pulls the actuating magnet from the transmitter. When this occurs, a silent radio signal is sent to the receiver which then sounds an audible alarm and activates any accessories attached to the receiver. There are two kinds of transmitters, a multi-use and a single use. The multi-use transmitter can be used up to ten times, depending on the length of time that the magnet is out of the transmitter shelf. The single-use transmitter is used only once then thrown out. You may monitor multiple births simultaneously, provided each expectant mare is wearing a transmitter and is within the range of the receiver.

When activated, the receiver will sound an audible alarm and activate any attached accessories. Both a long range and a standard range receiver are available. The long range receiver has dual antenna ports, allowing for monitoring both stall and paddock areas simultaneously. The practical range is 1000 to 1200 feet. The standard range receiver is typically used to monitor stall births. The practical range is 150 to 200 feet. Foal-Alert recommends that the receiver be placed as close to the birthing area, with as few barriers between the receiver and transmitter as possible. Barriers, particularly metal, may cause interference, thus reducing the range of transmission. In cases where an attendant is not nearby at all times, it is suggested that an accessory be used with the system. Available accessories include an auto-dialler or auxiliary alarm [FA17].

Cost

Long-range Foalert system to monitor 1 mare: €932.08

Long-range Foalert system to monitor 1 mare: €1440.98

Additional multi-use transmitter: €144.04

Additional single-use transmitter: €52.53

Advantages

- There are little chances of false alarms; unless the mare rubs the wearable off, only the delivery of the foal will signal an alarm.
- Scalable.

Disadvantages

- The wearable is significantly invasive. A veterinarian or competently trained person must suture the wearable on to the mare, and therefore the user will encounter additional costs (veterinary or special training).
- The system relies on radio frequency to activate the pager.
- A veterinarian must suture the device.
- The system relies on normal foal presentation to activate the alarm. If there is a dystocia the alarm may not be activated.

User Remarks

I used the Foal-Alert system for one year while working in the USA. We would attach the transmitter approximately five days before the anticipated due date. This was a significant drawback of using the system. Due to the invasive nature of the transmitter it was important to minimise the amount of time it was attached to the mare. As mares approach parturition and begin to develop an udder they become increasingly uncomfortable and irritated. So much so that a few individuals would rub the transmitter out by scratching against gates, fences, etc. Mares are inherently unpredictable when it comes to estimating their due date, and so attaching the transmitter at the correct time was extremely difficult. Attaching the

transmitter too earlier would increase the chances of the mare removing the transmitter resulting in an unusable transmitter and possible injury to the horse. While on the other hand you risk the mare foaling before you have the transmitter in place.

I didn't experience any difficulties with communication between the components as with the Equipage systems, perhaps because the repeater and pager were in close proximity to one another. Although the same limitations, regarding remote notifications, as discussed with the Equipage system also applies to the Foal-Alert.

The Foal-Alert system was not used as the first line of defence in ensuring an attended delivery, but rather as a safety precaution if one of the night-watchers were to accidently fall asleep.

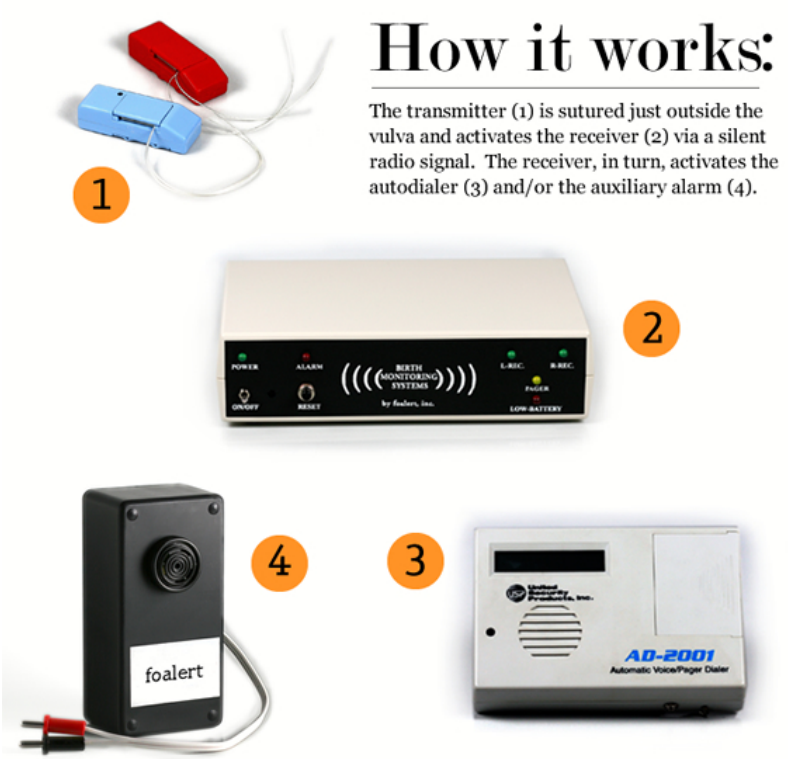


Image courtesy of Foalert. [FA17]

Foal App

The Foal App was released in July 2017. It is an Android/iOS application downloaded to two smartphones. One smartphone acts as the transmitter. A material pouch is provided to attach the smartphone to the underside of the mare's head on to her halter. The app utilizes the phones existing hardware (tilt sensor) to detect lying down. Foaling events will trigger a notification being sent to the second smartphone via Wi-Fi or mobile network. The user can activate the camera of the transmitting smartphone for real-time video monitoring.

Foal App is free to download and use as a receiver, however to use it to monitor a mare you must purchase credit within the app. This credit is per week, but does not expire until it is assigned to a mare and consumed. Foal App uses approximately 2GB of data per month on the monitor phones. This can vary depending on how often the end user chooses to access the live video feed.

Cost

Foal App pouch: €42.33

Weekly credits: €21.99

Advantages:

- The transmitter is non-invasive; it does not require a veterinarian or special training to attach to the mare.
- Installation and set-up of both systems are relatively simple.
- The transmitter is very low profile.
- Remote notifications via internet.
-

Disadvantages

- A smartphone that can run iOS 10.3.3 or Android 5.0 respectively is required for each mare for use as the transmitter.
- The device relies on WiFi or mobile network. Continuous wireless data use can quickly deplete device battery.
- Camera viewpoint is limited to the point of being non-functional.

Images courtesy of Foal App. [FP17]

Nightwatch Equine Distress and Wellness Monitor

Nightwatch is a smart halter designed to alert the end user via text, phone call, and/or email at the early signs of equine distress, such as colic or being cast. It is a patent-protected device that monitors real-time data on the horse's vital signs and behaviours. According to Protequus the device 'adapts to your horse over time as the system learns their unique and normal patterns and parameters' [NW17].

It communicates with a web application via WiFi or mobile network.

The device has the ability to measure heart rate, respiratory rate, activity, motion and position. The halter contains sensors, processors, transceivers and rechargeable batteries. It us enclosed in a water and dust resistant enclosure build in to the halter poll.

Cost

Nigthwatch halter to monitor one horse: €720.67

Advantages

- Ability to monitor horse's vital signs in real-time.
- The wearable is non-invasive; it does not require a veterinarian or special training to attach to the mare and is low reasonably low profile.
- Installation and set-up of both systems are relatively simple.
- Remote notifications via internet.

Disadvantages

- Non-scalable.
- Not suitable for all breeds. The halter is available in three sizes cob, standard and oversized, eliminating any horse under 14.2hh (average) from the target market.
- Cost. There is a substantial purchase price of each halter.

Images courtesy of Nightwatch. [NW17]

Notion

Notion is a home awareness kit that began as a Kickstarter project developed by Loop Labs Inc. The system is built around a hub and sensors which you place around your home. The hub is wireless and requires no hardwired Ethernet to connect to a home router. Setup involves plugging the hub into a power outlet and following setup instructions on an iOS or Android application.

The sensors are no larger than a hockey-puck and attach to surfaces with adhesive tape. Each sensor has the capability of measuring position change, temperature fluctuations, water leaks light and noise. Setup is relatively straight forward by scanning a QR code on the reverse of the sensor and completing setup via the mobile app. Functionality of each sensor can be enabled or disabled depending on the senor location and what the user intends to monitor.

Costs

Notions kit (three sensors, one hub): €185.62

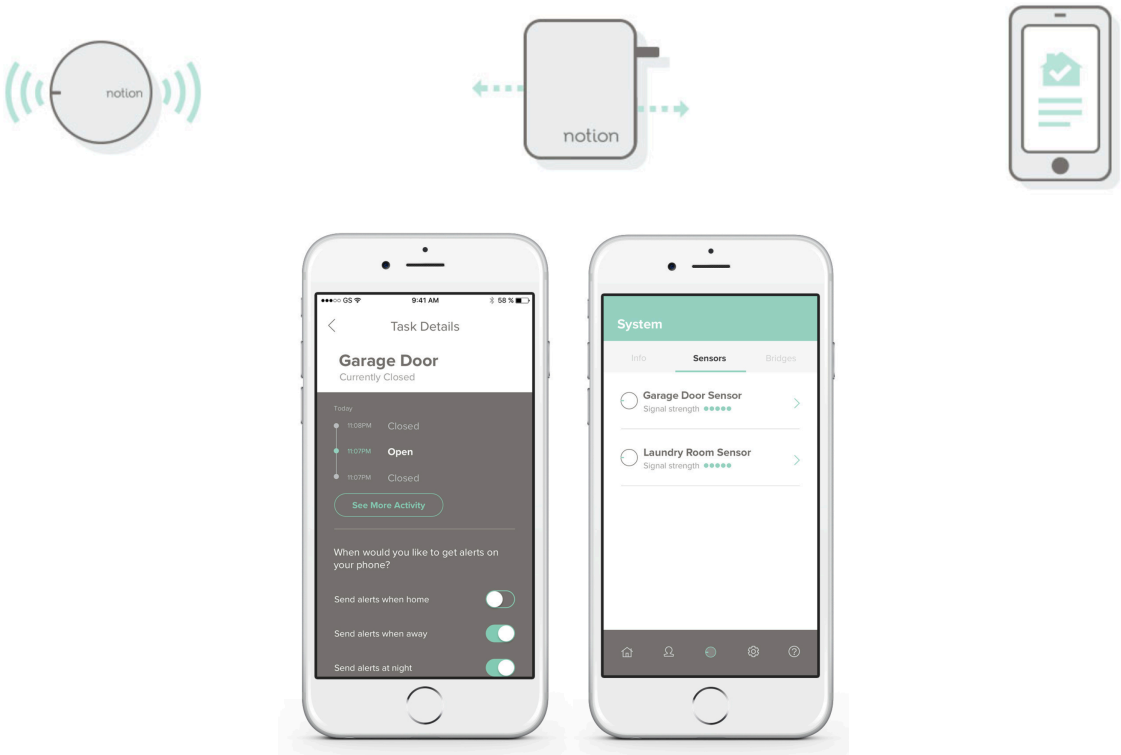
Additional sensor: €41.53

Advantages

- Ability to monitor multiple senses and enable/disable functionality of each sensor.
- Sensors and hub are low profile.
- Setup is relatively straight forward and user friendly.
- Remote notifications via Ethernet.
- Affordable and scalable.

Disadvantages

- Does not integrate with existing smart home component or systems.
- Alerts are limited to app pop-ups and are too easy to miss [CN17].



Images courtesy of Notion. [NN17]

Fitbark (Fitbit)

The Fitbark dog activity monitor is a canine activity sensor that allow dog owners to monitor their dog's daily activity. It is a canine version of the FitBit activity tracker. The device uses a 3-axis accelerometer to capture position and movement, and coupled with the species specific algorithms owners can track the time their pet spends playing, active or sleeping.

The plastic unit weighs a mere 0.28 ounces and measures 1.1 by 1.6 by 0.4 inches. It is waterproof as long as the rubber plug covering the micro USB charging port remains securely sealed. The Fitbark uses the micro USB connection to charge a lithium-ion battery which is stated to last for 14 days.

The system uses Bluetooth to transfer the recorded data to smartphone app. A WiFi base station is available separately for remote monitoring, which pushes the senor data to a cloud store.. The data is available to owners as in app statistics and analysis after 24 hours.

Similar to FitBit, the FitBark API enables developers to integrate FitBark datasets into third party mobile and web applications. Both Fibit and FitBark API uses the OAuth 2.0 protocol to authenticate incoming requests, which returns an access token with a user-defined-duration access token. The developer can make requests on behalf of the user using the access token. Request are returned as JSON objects from the Fitbit or FitBark API endpoints [FB17].

Cost

Fitbark dog activity monitor: €59.11

Fitbark WiFi Base Station: €50.68

Advantages

- Easy-to-understand activity information.
- Very light and unobtrusive.
- Attaches to any collar/harness.
- Long battery life.
- Waterproof.
- Web-based dashboard.
- Social aspect helps compare canines.

Disadvantages

- Lacks Wi-Fi, so you must buy a base station for remote monitoring.
- No real-time data

Image courtesy of Fitbark. [FB17]

Conclusion

There are multiple options available in today's market for foaling prediction and mare monitoring. The advantages and disadvantage of each device could be debated, and each system does attempt to address a significant limitation of the other yet none seem to encapsulate functionality, usability and scalability in a single cost effective system.

The Equipage and Breeder Alert do offer a simple, non-invasive system that can scale from the demands of the hobby breeder to that of professional horse breeders at a relatively low cost. Its mechanical build ensures easy installation, set-up and repair. However, its simplicity does result in significant limitations such as no remote monitoring; a dependency on RF communication; and, a rather verbose method of identifying imminent foaling.

The Foal App, does bring the consumer appeal of mobile technology to the equine breeding industry. For breeders with a small number of mares and existing network availability this could be a cost effective solution. However, forefront requirements include an Android or iOS smartphone with the latest version of respective operating systems and persistent wireless internet for each individual mare. Also, key selling points present significant usability and functionality concerns: such as relying on the phones camera for remote viewing; and the constant power requirements of continuous app use and internet connectivity. This systems dependencies greatly weaken its function.

The NightWatch system appears to be a leap forward in equine monitoring. I think it can safely be referred to as the Fitbit equivalent for horses with perhaps more advanced artificial intelligence. In an ideal world I believe this is the solution to not only foaling prediction but also the monitoring of breeding and competition horses for fitness, disease and stress. Unfortunately, its price tag and sizing greatly restricts its scalability, usability and target market respectively.

From my experience in using some of the above systems and from my research into existing products I feel that a suitable technological solution will bring together functionalities from each; such as the scalability of Equipage/Breeder Alert, the intelligence and functionality of NightWatch with the usability and consumer appeal of Foal App.

FOALING PREDICTORS

Before I can consider the hardware solutions to build the foaling alarm it is essential to understand the biometric parameters of which can be measured and their correlation to imminent foaling. A 1988 study on body temperature and the behaviour of mares during the last two weeks of pregnancy, carried out by Eve B. Shaw, Dorethy F. Holmes and Katherine A. Houpt, identified interesting findings that suggest a measurable pattern of mare behaviour and body temperature fluctuations in the days approaching delivery.

To study body temperature fluctuations, 19 pony mares where sampled twice a day for core body temperatures. The studied identified a significant drop, an average of 0.1 degrees Celsius, was observed the day prior to foaling. This would suggest that core body temperature could be considered as a potential signal of parturition and measurable biometric.

> *"Body temperature measurement has been described as a means of predicting onset labour in mares. A decrease in body temperature may begin about 4 hours prior to foaling." [PM09]*

While investigating mare behaviour, the team focused on 58 Thoroughbred and Standardbred mares during their last two weeks of pregnancy. It was observed that each mare spent an average 66.8% of their time standing, 27% eating, 4.9% lying in sternal recumbency, 1% lying in lateral recumbency, and 0.3% walking. The night before foaling mares were observed to spend a significant less amount of time lying in sternal recumbency. Their unlikely hood to 'settle' would suggest heightened levels of discomfort. On the night of foaling all behavioural patterns except that of eating were substantially different to the previous two weeks. There was a significant increase in walking, from 0.3% to 5.3%, as well as lying in lateral recumbency, increasing from 1% to 5.3%.

Of the 58 pregnancies the study observed 97% of mares foaled in a recumbent position [SH88].

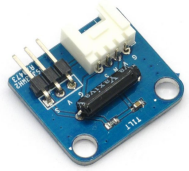
The observations of this study strongly suggest that position is a compelling predictor of imminent foaling.

HARDWARE

Sensors

The sensors will provide the mechanism for measuring mare biometrics. I will discuss the available sensor technology that could be used to build a wearable transmitter that can potentially capture the foaling predictors detailed in the previous section.

Tilt Sensor



The tilt sensor is a component that can detect the tilting of an object in two axis of a reference plane. They measure the tilting position with reference to gravity presenting an environmentally friendly alternative to the traditional mercury switch. They are comprised of a metallic ball that will commute the two pins of the device to 'on' or 'off' and vice versa when the sensor reaches a certain angle [AG17].

A tilt sensor would present the most basic mechanism for identifying a mare in lateral recumbency. However, the simplicity of a tilt sensor (on or off) presents an ambiguous means of identifying onset foaling and so its limitations may present issues, for example if a mare was foraging the forward-downward position of her head may trigger false alarms. To avoid this, we will need to determine the direction in which the mare's head is tilting be it left, right or forward (backward tilting would be a physical impossibility).

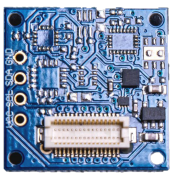
Temperature

Digital Non-Contact Infrared Thermometer

Infrared (IR) sensors measure the light bouncing off remote objects so that it can sense temperature without having to physically touch the object. The sensor is pointed at the object and will measure its temperature by absorbing the infrared waves emitted. More specifically the Sparkfun MLX90614 is contactless sensor capable of sensing object temperatures between -70 and 380 degrees Celsius. It is available in two levels of accuracy, 0.5 degrees Celsius and 0.1 degrees Celsius respectively. It has a measurement resolution of 0.02 degrees Celsius. The Sparkfun sensor produces two readings: the object temperature is the non-contact measurement using IR; the ambient temperature is the temperature reading on the die of the sensor.

Considering the findings from the 1988 study [SH88], temperature is a significant measure in predicting imminent foaling. The wearable design will directly affect the ability to accurately read temperature, consequently designing and positioning the wearable will be fundamental to accurately reading the horse's body temperature. Areas of minimal fat deposit, such as the horses face may provide the most accurate and stable body temperature reading. Positioning a wearable close to the horse's jaw or behind the ears may provide the most precise measurement while remaining minimally invasive.

Accelerometer



An accelerometer is an electromechanical device that will measure acceleration forces, which is the rate of change in velocity of an object. These forces may be static, such as gravity; or dynamic, such as movement or vibrations. Accelerometers can measure orientation with respect to the earth by measuring the amount of static acceleration due to gravity. The way in which the device is moving can be quantified by measuring the amount of dynamic acceleration.

Accelerometers are used to measure movement and activity in human/animal health trackers, so I believe this sensor will be an essential component of the wearable. Accelerometers produce readings for x, y and z axes. Using these outputs, we can determine if the mare is in lateral recumbency but also measure movement to identify the activities in the aforementioned study [SH88], such as walking which is confirmed to be a significant indicator of impending foaling.

Communication

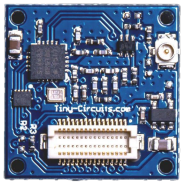
In order to exchange sensor measurements from the transmitter to the end user communication protocols need to be considered.

Wireless Communication

Wireless communication refers to the transfer of data between devices that are not connected by an electrical conductor. Wireless communication is an all-encompassing term to describe the many communication protocols for exchanging data between two or more devices using a wireless signal. Such protocols include: radio frequency (RF), an inexpensive means of long-distance communication for low cost application; cellular (GSM) for telephone systems, voice communication and data collection; Bluetooth/Bluetooth Low Energy for low-power, short range applications; and, WiFi for wireless data communication.

Radio Frequency

One of the easiest and inexpensive ways of implementing wireless communication is using radio frequency (RF). A RF module is a cheap wireless communication module for low cost application. A module consists of a transmitter and receiver that operate at the same radio frequency.



Data transfer via RF can be characterised in three stages:

- First, there is a transmitter which begins the RF communication. The transmitter takes the initial data and modifies the signal using a modulation technique to encode the data into the signal.
- Next, an antenna collects the signal that it receives from the transmitter and directs the RF waves away from the antenna. As the RF waves move away from the transmitting antenna they move towards another antenna attached to the receiver.
- Finally, the receiver takes the signal that it received from the antenna and translates the modulated signals and passes them on to be processed [AG15].

RF communication in its simplest form; using single transmitter and receiver is a one-way communication protocol. The transmitter broadcasts at a predefined frequency, then the receiver is listening on that frequency. To achieve two-way communication, for example confirming that an alarm event was received by the hub, would require transceivers.

The range of microcontroller RF transmitters/transceivers is approximately 500 metres. This range is theoretical and via unobstructed line of sight. Buildings, trees and devices operating on the same frequency could potentially obstruct communication and decrease its range. Although depending on the end users' location and environment RF communication between the wearables and the central hub would offer the highest degree of flexibility on where to position the central hub and whether or not Ethernet access is required in the horse barn/stables.

Bluetooth

Bluetooth is a standardized protocol for sending and receiving data over a 2.4GHz wireless link. It's a secure protocol, and it's perfect for short-range, low-power, low-cost, wireless transmissions between electronic devices [JM13]. A Bluetooth piconet (a Bluetooth network) uses a master-slave model to control the transfer of data. A single master device can pair with up to 7 devices (although 3 or 4 is the practical limit). Each slave device is connected to the master and can communicate to that device only.

Traditional Bluetooth has a theoretical range of 100 metres. The connecting process involves one time pairing of the devices and usually an authentication step (v2.0 and earlier). Both devices share a common secret key which allows them to automatically par in the future.

Bluetooth Low Energy (BLE) was introduced in Bluetooth v4.0. BLE completely changed the communication protocol from master/slave to broadcast/subscription. BLE requires significantly less power over tradition Bluetooth, but to do so sacrifices range (theoretical range of 50 metres) and data throughput (0.27 Mbps instead of 0.7-2.1 Mbps). BLE was designed for peripheral battery dependent devices such as smart watches and fitness trackers [JM13].

WiFi

WiFi is one of the most common communication protocols. It uses an infrastructure network of communication to establish a bridge to other networks, access control and forwarding. The access point (router) handles the network functions so that clients (connected devices) remain simple. WiFi is a star based network in that communication is established between wireless nodes to a wireless access point.

WiFi range is effected by multiple factors such as: the current standard running on the device, later standards have greater range over earlier ones; and, physical obstacles have a significant interference with range. WiFi typically has decent range coverage and can penetrate obstructions such as walls. Also, adding and removing devices in the network is relatively simple. However, in contrast to previously mentioned communication protocols WIFI has the highest power demand and device proximity to the access point (router) is similar to that of BLE.

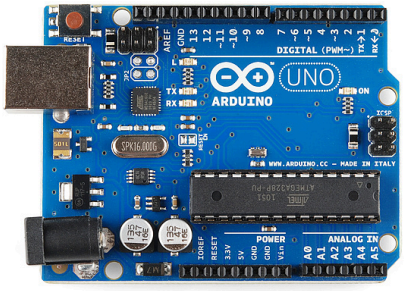
Considering the energy and possible size requirements of adding WiFi capabilities to each wearable, I believe that RF transceivers do bear an advantage for the scope of this project.

WiFi (as well as wired Ethernet) should be an available option for connecting the central hub to the cloud backend.

Controllers

The controller or central hub will be the communication access point between the wearable foal alarms and the end user. The central hub will communicate foaling alarm events to the connected users via a cloud backend service. Requirements include internet connectivity, scalable RF transceiver/receiver connectivity, ability to run concurrent code.

Arduino



Arduino is an open-source platform for building electronic projects [BN13]. The Arduino platform consists of a physical programmable microcontroller (circuit board) and an integrated development environment to write and upload code to the physical board. The Arduino IDE uses a simplified version of C++ [BN13].

Arduino can interact with a multitude of hardware components such as buttons, LEDs, motors, speakers, GPS units and cameras; as well as a variety of environmental sensors over a range of communication protocols.

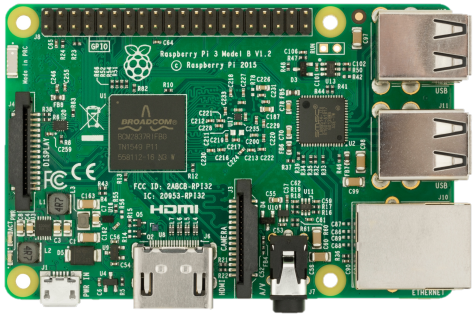
The Arduino compiler accepts both C and C++. The Arduino IDE generates function prototypes and then passes the code to a C/C++ compiler. Arduino is a single-core chip with procedural code and so is not capable of multi-threading. They simply execute code as their firmware interprets it. Considering the requirements of this project, where by potentially several wearable foaling alarms may be simultaneously communicating with a central hub, the code must be able to handle concurrent data transfers from the

connected wearables. To overcome this, we can implement protothreading as a means of achieving concurrent programming on Arduino.

The Arduino environment and platform is covered under the GPL license. This requires: any circuits that are made as derivative of the Arduino board must release design files with a CC-BY-AA license; any modifications to core-files must be made available to everybody as well as all object files that allow for re-linking of the firmware against updated versions of the Arduino core libraries [AR16].

Raspberry Pi

A Raspberry Pi is a general-purpose computer, usually with a Linux operating system, and the ability to run multiple programs [PJ15]. It has all the attributes of a computer, with a dedicated processor, memory and graphics driver.



As the Pi essentially is a computer it has the ability to execute python code and therefore facilitate concurrent programming. If we are to consider streaming, for example, accelerometer data for the wearable foal alarm to identify mare activity patterns a Raspberry Pi would be a preferable choice of hardware to use as the central hub.

It's advantages in functionality does make it a more complicated component. The Arduino executes code procedurally as its firmware interprets it. In the event of a restart (e.g. due to power loss) the Arduino will simply continue to interpret and execute its code without the need of intervention by the end user. The Pi is for all intents and purposes a full-fledged computer. Due to its complexity if would require end-user assistance in such an event.

Requirements

Arduino microcontrollers, specifically Arduino Uno WiFi for use as the main controller or hub and Tinyduino for use as the wearable foaling alarm, meet the needs of this project. Tinyduino is a platform based on Arduino technology. They offer processor boards and sensors at a fraction of the size of the standard Arduino. Additionally, the Tinyduino boards are stackable so do not require soldering. The platform is aimed at building discrete wearables and sensor devices, and offers a stackable accelerometer, temperature sensor, RF transceiver, USB interface board and rechargeable lithium ion battery [TC17].

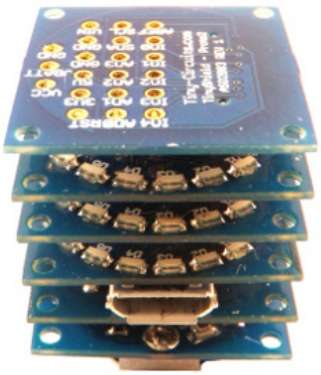


Image courtesy of Tinycircuits. [TC17]

DATABASE

A database is a software application that captures data in some organized way and facilitates accessing, managing or updating of the data by end users and other applications. In general, a database management system (DBMS) will allow the definition, creation, update and querying of data to derive information. DBMS can be classified according to the data models they support.

Databases Types

For the purpose of research for this project two primary types of database models will be considered: SQL, or relational databases; and, NoSQL databases.

SQL

Structured query language (SQL) is the standard application interface to relational database management systems. Relational databases adhere to a strict schema organized in a tabular fashion, presenting the data and information with rows and columns. A table is a collection of objects of the same type, i.e. rows, and can be referred to as a relation. The columns represent attributes of the object (or row).

"Relational tables follow certain integrity rules to ensure that the data they contain stay accurate and are always accessible." [RD17]

The rows in relational database tables must be distinct, discernible by a unique identifier referred to a primary key. Another integrity rule requires that columns may not be repeating groups or arrays of an attribute. Where there is a need for repeating attributes, those attributes may be represented in a relation (a row) to that particular row. Following this schema, the programmer can define their data models according to relationships between the data objects.

A relational database must adhere to ACID properties, which are: atomicity, where an operation must be complete and indivisible; consistency, data is only committed if correct in type while adhering to triggers and constraints; isolation, meaning one transaction won't affect another; and durability, referring to in the event of a failure or crash data and transactions will remain persistent.

Relational databases are best suited to structured data and applications that require high-performance transactional workloads.

NoSQL

'Not Only SQL' is an alternative means of data storage and retrieval other than the tabular relations of SQL databases. NoSQL databases may refer to object, document, key/value or graph datastore. They do not adhere to the structure, consistency and predefined schema of a relational database model and, have no declarative query language, whereas database operations resemble functions rather than standard SQL.

NoSQL databases aim for eventual consistency, and adhere to CAP properties which state: consistency, all distributed replicas of the data must be the same value; available, meaning every request will receive a response without guarantee; and partition tolerance, meaning that the system is designed to continue operating in the event of a partition failure or server loss. Of the three CAP properties only two can be offered by a system at one time.

NoSQL databases do offer some advantages over traditional relational models. The lack of structure or schema allows for frequent changes and the addition of varied data objects. Furthermore, they are scalable, designed with 'Big Data' in mind and run well in a distributed environment i.e. the cloud.

Google Firebase Real-Time DB

Firebase is Google's web application development platform. It started as a backend as a service (BAAS) but as of mid 2016 it has been comprehensively extended to become a complete backend solution for web and mobile application development. The platform has one SDK and one console to create and manage applications for web, iOS and Android devices [AA16].

The Firebase real-time database is a cloud NoSQL database.

> *"Store and sync data with our NoSQL cloud database. Data is synced across all clients in realtime, and remains available when your app goes offline." [FB17]*

The data is stored as JSON objects and is synchronized to every connected client. Updates are automatically pushed in realtime to every connected app. All connected clients share a single instance of the Firebase realtime database. Instead of using HTTP requests Firebase uses data synchronization to automatically update every connected client when a database change occurs. The database is accessible directly from a web or mobile application without the need for backend services. The database utilizes an expression-based rules language to define how the data is structured and user permissions. When used in conjunction with Firebase authentication services the programmer can determine who can access the data, the type of data they can access and the operations an individual user may perform on the data.

Requirements

Realtime notifications are essential to this projects functionality and success. Foaling alarm events will be pushed to a cloud-based database, upon which all connected clients, being instances of a web application, a mobile application or mobile phones (texts) must be notified upon any changes to the database. Firebase's realtime database provides the necessary functionality without the need for additional services to synchronize data and automatically push database changes to connected clients and SMS or email notification handlers.

Node

Node.js is an open-source, cross platform JavaScript runtime environment. It follows an asynchronous and event driven input\output model that executes server-side JavaScript code.

> *"Node.js eliminates the waiting, and simply continues with the next request. Node.js runs single-threaded, non-blocking, asynchronously programming, which is very memory efficient." [W317]*

It provides a means of using a traditionally client side scripting language on the backend, dynamically generating web content before the pages are sent to the browser.

NPM (Node Package Manager) is a package manager for the JavaScript programming language. It is a command line tool to install and update dependencies associated with Node projects. It makes code reusable while reducing project development time by abstracting common and generally complex tasks into packages that can be installed locally or globally to the project.

Google Cloud Functions

Cloud computing has made possible serverless application architecture. Traditional sever-side logic is replaced by micro services that may be invoked on-demand in response to a multitude of events originating from anywhere, and most importantly without the need for hardware infrastructure.

Google Cloud Functions is a serverless environment that enables the programmer to build cloud-based backend services at the level of a single function [GC17]. Event-driven code can trigger logic to listen or respond to actions.

> *"Each Cloud Function runs in its own isolated secure execution context, scales automatically, and has a lifecycle independent from other functions." [GC17]*

'Cloud Functions offers a Node runtime environment to build cloud services in JavaScript while Google manages infrastructure, operating systems and VMs on the developer's behalf. Firebase natively emits events which when coupled with Cloud Functions and other Google cloud services such as datastore, authentication and analytics can totally remove the need for traditional server infrastructure.

Requirements

Google cloud Functions, which provides a Node runtime environment, will replace the need for the standard client-server model of application development for this project. Server-side logic such as sending SMS and email notifications can be built as a cloud service and triggered as a response to changes in the database. Server-side authentication and data validation will be handled by Google Firebase Authentication which provided functions of authenticating clients by email, password, telephone or several third party applications such as Facebook or GitHub. Simple abstracting the logic of authentication, password validation and user management allowing the developer to focus on the main functionality of the project. Data validation will consist of both client and server-side data validation using Angular validators and Google Firebase expression-based rules language. As a result of my research I believe that server-side logic for this particular project can be delegated to cloud services.

APP DEVELOPMENT

The user application must provide a means of user and hub registration and authentication, viewing real-time foaling alarm events, managing wearable foaling alarms, viewing connected IP cameras and crowd sourcing mare monitoring. I considered and researched native and web based application development for this project.

Native

Native application development refers to the development of iOS or Android mobile apps using Swift or Java respectively. Performance is the main advantage of native application development. Native applications persist on the mobile device, are platform specific, compiles into machine code and have access to the full hardware and software functionality of the device.

There is a significant learning curve associated with native application development as the code must be written for each specific platform. Although logic may be the same, the programming language, APIs and the development process is different amongst iOS and Android platforms. As a result, little of the code will be able to be shared or reused.

Web

Web applications run in the browse. They are platform independent and can be viewed on any device with an internet connection and web browsing capabilities. There exists a plethora of frameworks for the development of web applications in the majority of programming languages such as PHP, Python, Ruby and JavaScript. Although fundamentally, all web apps are displayed using HTML5, CSS and JavaScript.

Angular is an open-source, front-end web application platform developed and maintained by Google. Released in 2016 it succeeded AngularJS as the next MV* framework for building complex web applications.

> *"AngularJS has become a popular framework in building Single Page Web Application. It is a Javascript web framework that was written with Testability in mind." [JM16]*

Angular replaces the AngularJS concept on controllers with components. Components allow developers to separate the functionality of the application into reusable, self-contained groups of modules. Modules being task specific logical pieces of code. Each component will have a template that defines the view, i.e. the HTML and styling. Components can communicate with each other through dependency injection where shared services can be used across multiple components or applications.

> *"In addition, Angular 2 has better event-handling capabilities, powerful templates, and better support for mobile devices." [TP17]*

Angular is Typescript based, which is a strictly-typed superset of JavaScript. The use of typescript allows for the development of large scale front-end applications that adhere to a strict syntax unlike vanilla JavaScript applications. Additionally, Typescript syntax may be relatively familiar to backend programmers, meaning those skills can be transferred to front-end development also.

Angular CLI is a command line interface to automate the development of Angular 2+ applications. It allows the developer to: quickly set-up and scaffold an Angular application; run a development server; generate components, routes, services or pipes and automatically integrate them into the project; and generate unit and end-to-end tests and test the application locally during development.

Requirements

Consideration of end user requirements will dictate the type of application and platform selected for the development of this project. The user must be able to register, authenticate, manage alarms and receive realtime notifications remotely. Further functionality may include viewing embedded IP cameras to monitor mares and the notion of crowd sourcing mare monitoring, where by foaling alarms and their respective IP cameras may be viewable by all logged in users in an attempt to increase the possibility of an attended delivery.

As no native functionality is required I feel that native app development is not justifiable. Also both platforms, iOS and Android, would need to be developed to facilitate all end users.

I believe a web application is essential for registration, alarm and mare management (CRUD operations), viewing IP cameras and crowd sourcing mare monitoring. Mare owners will typically rely on camera monitoring to ensure an attended delivery, viewing the cameras via television, computer or tablet. By embedding the camera in users' web application console we can provide a platform that converges both visual and sensor monitoring.

PROGRAMMING LANGUAGES

Arduino

The Arduino programming language is nothing more than a subset of C++ functions that are designed to abstract complexity while making programming simple and easy for beginners. The Arduino compiler can accept C++ or C code.

C++ is a mid-level programming language developed by Bjarne Stroustrup in 1979 [CP17]. The language can run on a variety of platforms such as Windows, Mac, several versions of Unix and Arduino.

Sensors are interfaced using libraries such as: VirtualWire [MM16], for RF transmitter-receiver communication; ArduinoFirebase [AF17], for calling the Firebase API from the Arduino core; and, Arduino-Ethernet [AE17], to allow the Arduino board to connect to the internet.

JavaScript/Typescript

JavaScript is the most commonly used client-side scripting language. It is lightweight, high-level, loosely-typed and multi-paradigm based. Together with HTML5 and CSS it is a core technology of web application development, with JavaScript facilitating user interaction, dynamic content, control and animation [MO17]. JavaScript can function as both a procedural and an object-oriented programming language.

> *"Objects are created programmatically… at run time, as opposed to the syntactic class definitions common in compiled languages like C++ and Java." [MO17]*

Typescript is a superset of JavaScript providing the programmer with a strictly-typed syntax, classes and interfaces. Therefore, common bugs associated with programming in JavaScript can be avoided with type-checked code and providing a richer environment and structure for spotting errors as code is developed.

> *TypeScript can actually report issues without you even saving your file, and leverage the type system to help you write code even faster." [DS17]*

Typescript was developed by Microsoft and also happens to be the language in which Google developed their new Angular framework. Typescript attempts to bridge the skills of the front-end developer whose knowledge is generally entrenched in JavaScript with those of the backend-developer that is familiar with strong-typed, compiled programming languages.

> *"It provides advanced autocompletion, navigation, and refactoring. Having such tools is almost a requirement for large projects." [VS16]*

As Typescript is not interpreted by browsers, to be executed it is compiled to JavaScript.

Conclusion

For the development of the hardware components and interfacing with sensors (accelerometer, RF transmitter/receiver or temperature sensor), the C++ programming language will be used in conjunction with Arduino libraries.

Google Cloud Functions provide a Node runtime environment and therefore JavaScript will be used to develop the projects cloud (backend) services.

As typescript is the primary language of choice for Angular, this will be used during the development of the web application.

DEVELOPMENT TOOLS

"As a developer, your code editor is one of the most important parts of your setup."
[HF17]

Development tools will depend on programming choices. Integrated development environments provide powerful tools for project creation and initialization; classes, interface and service generation; testing; and debugging. I have chosen to research the most popular and available IDE with respect to the frameworks and programming languages previously mentioned.

Visual Studio Code

Visual Studio Code is a code editor developed by Microsoft for Windows, Mac and Linux platforms. I combine the ease of use of a lightweight editor yet has support for debugging, version control and integrated terminal. It can be extended into a powerful IDE with the multitude of third party extensions, resulting in a fully customizable development environment for the individual.

VSCode provides IntelliSense and linting by default for JavaScript, Typescript, JSON, HTML, CSS, Less, and Sass [HF17]. Additionally, plugins and language support extensions are suggested during coding, rather than the developer having to search for the correct install.

Webstorm

Webstorm is a cross platform IDE primarily for JavaScript and Typescript development. It is developed by JetBrains and is available as a free download to students. It has a built in web server that allows the developer to run and test projects from the integrated terminal. The IDE has a local history feature that tracks all local changes made to code and allows the programmer to roll back changes whenever necessary. It comes bundled with linters for JavaScript and Typescript. Additional plugins can be downloaded via NPM. The IDE also includes support for React, Meteor and Angular, allowing developers to quickly scaffold a project in their desired framework and automatically install dependencies and generate directories. With integrated version control for Git, developers can make use of Webstorms visual tool for commits, push/pull, merging, etc. Additionally, GitHub is available natively allowing developers to directly check out projects from remote repositories.

Arduino IDE

The Arduino Integrated Development Environment contains a code editor, message area, integrated console and a toolbar for common functions. It connects to the Arduino or Genuino hardware to upload program sketches and interface with them [SM15].

Programs written in the Arduino IDE are referred to as sketches. The sketches may be written in the C, C++ or the Arduino language (which is simply a subset of C functions) then saved as a .ino file.

The target board is configured through the 'tools' menu. The programmer selects the board as identified by the computer's USB port and the port – the serial device, real or virtual connected to the computer. Typically, the programmer will verify and compile the code before uploading it to the board. The IDE's compile function checks for programming errors, memory usage and displays it in the integrated console. The upload functionality compiles and loads the file(s) to the board.

DEPLOYMENT

Google Firebase

Firebase hosting is production grade hosting for developers. It is developed for mission-critical websites and apps with a focus on security, scalability and reliability [CR14]. Core features of Google's Firebase hosting include: CDN-backed global content delivery; automatic SSL certificate provisioning; support for non-SNI devices, ensuring all users have a secure experience; and, custom domain support [CR14]. No other hosting service matches these features, states Google developer Chris Raynor.

Firebase hosting is aimed at dynamic applications. Traditionally, web applications dynamically generated HMTL on the server every time a page was requested. This was slow, difficult to scale and required the user to connect to servers that may have been a great distance from their location to load content. This model required client-server communication on every state change, and for mobile clients, applications would break when connection was lost.

With the rise of JavaScript frameworks, faster JS engines and more powerful mobile devices client-side content generation has become the norm. Frameworks like Angular have changed the way in which content is delivered to end users. Rather than servers sending content to each client a single optimized set of static assets can be served to all connected clients [CR14]. These static assets can be delivered over a geographically distributed CDN instead of a distant server.

Firebase eliminates the need for server-side code and allows applications to update data in realtime. Google Cloud functions allow the developer to perform server-side processing.

"The Firebase CLI provides tools for managing viewing and deploying firebase projects." [FB17]

The Firebase command line enables the developer to install Firebase-Tools to quickly and seamlessly get a projects to production. With a simple command Firebase can build your project, initialize the directory structure and generate configuration files. Then finally the deploy command, by default, creates new releases of all deployable resources in the project [FB17].

Firebase pricing is divided into Spark, Flame and Blaze plans. Spark is free, offering 10 GB/month bandwidth, up-to 100 simultaneous connections and 5 GB cloud storage, realtime database and cloud functions. Flame pricing is fixed at $25 per month aimed at new and growing apps. While Blaze pricing scales depending on usage and requirements.

Heroku

Heroku, released in 2007, is a cloud platform as a service (PAAS) with support for most web application development languages, such as Node, Python, Ruby, Scala, PHP and Go. Heroku allows developers deploy projects with minimal commands using common developer tools like Git, GitHub and Docker. Heroku is described as a polyglot platform meaning programming language is reduced to syntax and libraries. Heroku deployment infrastructure means that deployment endpoints and models are language agnostic [AW11].

Heroku aims to abstract the complexity of project deployment, configuration, managing dependencies and infrastructure so that the developer can focus on the application's functionality.

"We'll build and run your application with our container based platform, handling compilation, dependencies, assets and executables so you can focus on creating engaging experiences for your users." [SF17]

Heroku price plans range from free to enterprise level. Pricing is considered in terms of dynos (a Heroku container which hosts the application) and processes, where project owners only pay for what they use. The free plan is aimed at experimentation, offers a single process which sleeps after 30-minutes of activity. Hosting for personal projects start at $7 per dyno/month.

Although Heroku can be relatively easy to start with, future deployments are slow. Larger applications may have slower deployments as it can time for dynos to restart. Furthermore, Heroku can become quite expensive outside of the free tier [EN17].

CONCLUSION

The use of sensors and wearable technology for animal health management is an emerging market that is becoming increasingly popular. The number of sensors being developed and the degree of functionality greatly varies with respect to industry. With some advanced technologies for precise health monitoring and disease diagnosis being confined to human applications. Gradually these technologies are being transferred to animal industries such as livestock production, equestrianism and companion animals. Biosensors and wearables are being considered in an effort to: refine livestock production practices and costs; improve equine training, conditioning and breeding techniques; and identify behavioural and health characteristics to improve the life of companion animals.

The use of sensor technology for foaling prediction has been available since 1987 [EP14]. These products utilize the mare's tendency to lie in lateral recumbency moments prior to and during foaling. Sensor technology can be limited to a tilt-sensor to identify the mare's position and a repeater/pager to alert the nearby attendant. It is undoubtedly a simple system, non-invasive for the mare and easy to install and use for the breeder. However, it's simplistic design limits its functionality. Firstly, correlating a single instance of lateral recumbency to immanent foaling is rather verbose. Additionally, there are cases when a mare's head position may falsely identify lateral recumbency to the tilt sensor such as foraging or rolling. Finally, the use of a repeater/pager to wirelessly communicate foaling alarm events greatly restricts the breeder's ability to continue their daily routine and monitor mares remotely.

Considering the 1988 study on behaviour of mares during the last two weeks of pregnancy by Shaw, Holmes and Houpt, we can conclude that both lying position and walking activity are measurable biometric factors in determining immanent foaling [SH88]. Using a 3 axis accelerometer to monitor mare locomotor activity during the last two weeks of pregnancy may successfully identify movement patterns to predict the onset of foaling. Furthermore, lying in lateral recumbency was concluded to be a significant behavioural trait of mares prior to and during foaling.

The proposed system would utilise accelerometer sensors and wearable technology to upload locomotor activity to a cloud server for processing and data analysis. Upon identification of foaling alarm events end users will be notified by sending a SMS or email alert. Finally, users will manage wearable foaling alarms and monitor expectant mares via a web application.

REFERENCE

[CK09] Charlotte Kirkpatrick. (10/03/09). *Foaling may be a case of life or death,* (online), available: http://www.independent.ie/business/farming/foaling-may-be-a-case-of-life-or-death-26519952.html [accessed: 14/10/17].

[PW16] Pam Wilkins, DVM, PhD. (16/02/16). *Progress in foal survival rates,* (online), available: https://equusmagazine.com/diseases/progress-foal-survival-rates-31363 [accessed: 14/10/17].

[PR12] PM McCue, RA Ferris. (01/02/2012). *Parturition, dystocia and foal survival: a retrospective study of 1047 births,* (online), available: https://www.ncbi.nlm.nih.gov/pubmed/22594021 [accessed: 14/10/17].

[EP14] Kee-Port Systems. (01/01/14). *Equipage Foaling Alarm*, (online), available: http://foalingalarm.com/equipage.htm [accessed: 18/10/17].

[FA17] Foalert. (01/01/2017). *Foalert Components,* (online), available: http://foalert.com/components/ [accessed: 18/10/17].

[FP17] Foal App. (01/07/17). Foal App: How it works, (online), available: https://www.foalapp.com/how-it-works [accessed: 18/10/17].

[NW17] Protequus. (01/01/17). *Nightwatch: Features and Benefits*, (online), available: http://www.nightwatch24.com/features#features-and-benefits [accessed: 18/10/17].

[CN17] Christopher Null. (24/04/2017). *Notion Home Awareness Starter Kit review: Smarter sensors, but limited control,* (online), available: https://www.techhive.com/article/3191786/sensors/notion-home-awareness-starter-kit-review-smarter-sensors-but-limited-control.html [accessed: 19/10/17].

[NN17] Notion. (01/01/17). *Get Notion: How it works*, (online), available: http://getnotion.com/how-it-works/ [accessed: 19/10/17].

[FB17] Fitbark. (01/01/17). *Fitbark Developer API,* (online), available: https://www.fitbark.com/dev/ [accessed: 19/10/17].

[SH88] E. Shaw, K. Houpt, D. Holmes. (02/05/1988). *Equine Veterinary Journal: Body temperature and behaviour of mares during the last two weeks of pregnancy,* (online), available: https://www.researchgate.net/publication/227891341_Body_temperature_and_behaviour_of_mares_during_the_last_two_weeks_of_pregnancy [accessed: 19/10/2017].

[PM09] Patrick McCue. (01/04/2009). *Prediction of Foaling*, (online), available: http://csu-cvmbs.colostate.edu/Documents/Learnmares32-pregfoal-prediction-apr09.pdf [accessed: 19/10/2017].

[AG17] Anders Gran. (01/01/2017). *Tilt Sensor,* (online), available: https://www.arduino.cc/en/Tutorial/TiltSensor [accessed: 19/10/2017].

[AG15] Alexandra Gunter. (30/06/15). *How are RF signals transmitted,* (online), available: https://boundless.aerohive.com/technology/wi-fi-back-to-basics-radio-frequency-rf-components.html [accessed: 19/10/17].

[JM13] Jimb0 (26/10/2013) *Bluetooth Basics,* (online), available: https://learn.sparkfun.com/tutorials/bluetooth-basics [accessed: 26/10/2017].

[BN13] B_E_N (26/02/2013) *What id Arduino?*, (online), available: https://learn.sparkfun.com/tutorials/what-is-an-arduino [accessed: 31/10/2017].

[AR16] Arduino (01/01/2016) *Arduino FAQ,* (online), available: https://www.arduino.cc/en/Main/FAQ#toc10 [accessed: 31/10/2017].

[PJ15] Patrick Di Justo (04/12/2015) *Raspberry Pi or Arduino,* (online), available: https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/ [accessed: 31/10/2017].

[TC17] Tinycircuits (01/01/2017) *Tinycircuits: Tinyduino*, (online), available: https://tinycircuits.com/ [accessed: 05/11/2017].

[RD17] Java Tutorials (01/01/2017*) A Relational Database Overview,* (online), available: https://docs.oracle.com/javase/tutorial/jdbc/overview/database.html [accessed: 05/11/2017].

[AA16] Abel Avram (20/05/2016*) Google Firebase – A complete backend solution for mobile and web,* (online), available: https://www.infoq.com/news/2016/05/google-firebase [accessed: 05/11/2017].

[FB17] Google (31/10/2017) *Firebase Realtime Database,* (online), available: https://firebase.google.com/docs/database/ [accessed: 05/11/2017].

[W317] W3 Schools (01/01/2017) *Node.js introduction,* (online), available: https://www.w3schools.com/nodejs/nodejs_intro.asp [accessed: 06/11/2017].

[GC17] Google Cloud Platform (31/11/2017) *Cloud Functions*, (online), available: https://cloud.google.com/functions/ [accessed: 06/11/2017].

[JM16] Jason Martajaya (07/06/2017) *AngularJS vs Angular 2, which one to choose?,* (online), available: https://magenic.com/thinking/angular-2-vs-angularjs-1-which-to-choose [accessed: 06/11/2017].

[TP17] Tutorials Point (01/01/2017) *Angular 2 Tutorial*, (online), available: https://www.tutorialspoint.com/angular2/index.htm [accessed: 06/11/2017].

[CP17] Tutorials Point (01/01/2017) C++ Tutorial, (online), available: https://www.tutorialspoint.com/cplusplus/ [accessed: 06/11/2017].

[MM16] Mike McCauley (26/03/2016) *Virtual Wire,* (online), available: http://www.airspayce.com/mikem/arduino/VirtualWire/ [accessed: 06/11/2017].

[AF17] Firebase (27/10/2017) *ArduinoFirebase*, (online), available: https://github.com/firebase/firebase-arduino/commits/master [accessed: 06/11/2017].

[AE17] Arduino (01/01/2017) *Arduino-Ethernet: Ethernet / Ethernet 2 library*, (online), available: https://www.arduino.cc/en/Reference/Ethernet [accessed: 05/11/2017].

[MO17] Morfioce (15/10/2017) *JavaScript, what is it?,* (online), available: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript [accessed: 05/11/2017].

[DS17] David Smooke (27/06/2017) *Typescript vs JavaScript*, (online), available: https://hackernoon.com/typescript-vs-javascript-b568bc4a4e58 [accessed: 05/11/2017].

[VS16] Victor Savkin (22/07/2016) *Angular: Why Typescript?,* (online), available: https://vsavkin.com/writing-angular-2-in-typescript-1fa77c78d8e8 [accessed: 05/11/2017].

[HF17] Hugo Di Francesco (07/02/2017) *Visual Studio Code: the editor I didn't think I needed*, (online), available: https://hackernoon.com/virtualstudio-code-the-editor-i-didnt-think-i-needed-16970c8356d5 [accessed: 06/11/2017].

[SM15] SM (09/07/2015) *Arduino Software IDE*, (online), available: https://www.arduino.cc/en/Guide/Environment [accessed: 06/11/2017].

[CR14] Chris Raynor (13/05/2014) Introducing Firebase Hosting, (online), available: https://firebase.googleblog.com/2014/05/introducing-firebase-hosting.html [accessed: 06/1/2017].

[FB17] Firebase (31/10/2017) Firebase CLI Reference, (online), available: https://firebase.google.com/docs/cli/ [accessed: 06/11/2017].

[SF17] Salesforce (01/01/2017*) Focus on your app not the infrastructure*, (online), available: https://developer.salesforce.com/platform/heroku [accessed: 07/11/2017].

[AW11] Adam Wiggins (03/08/2011) *Polyglot Platform,* (online), available: https://blog.heroku.com/polyglot_platform [accessed: 07/11/2017].

[EN17] Endi (29/10/2017) *Slant: What are the best hosting services for Node.js apps?*, (online), available: https://www.slant.co/topics/250/~best-hosting-services-for-node-js-apps [accessed: 07/11/2017].