AnimalAware

Group 28: Arsalan Zahid, Purv Patel, Ian, Zeel Patel

Project Description

AnimalAware is a large-scale database that allows zookeepers to keep track of their Animals' health and mood, including sending alerts to them as necessary, and providing an interface for staff to interact with each other and send each other information related to the animals' health. This allows the zoo keepers to ensure the animals are healthy, and makes it much more efficient compared to zoo keepers previously trying to run to each other in person. Additionally, the system provides emergency alerts whenever needed, ensuring that the animals always stay healthy, even if zookeepers do not watch them constantly.

The Client's business is a zoo, which takes in endangered and injured animals for preservation purposes. The zoo handles large animals and attempts to nurse them back to health. Currently, the main method of observing animals is manual and direct observation of the animals by the zookeepers and the staff. As of right now there are no other competing products that can do such a thing. This is likely due to the large research overhead required for a product like this. At the moment, the closest alternative would be a live feed of CCTV cameras that the staff would have live access to; however, that would not provide alerts on the animals' health such as their mood or their vaccination status.

An example of how this product could be used is Zookeeper Billy is working his regular shift at Dallas Zoo when he notices Ellie the elephant has been sleeping longer than usual. He opens AnimalAware on his computer, and then sees a general trend of depression for Ellie the elephant. To ensure everything is okay, he sends a request for her to be checked by the zoological veterinarians through AnimalAware, which is then approved by the zoo managers.

Requirements

Because delays in critical alerts could endanger an animal's health, the product is required to have low latency. Alerts must reach the user within 45 seconds, and all queries shall take less than 100 milliseconds. The model shall not have any false negatives, because that could mean critical events go unnoticed by the zoo keepers. Additionally, the application should be able to handle thousands of users working at the same time, and be able to easily transfer data between the user and the server. At the same time, as the application is handling thousands of users, it must not fail any of the previous requirements, or else the safety of animals is at risk.

Additionally, security is one of the highest priorities, as an insecure system may be a prize for ransomware attacks, which hurts the reliability of the system. For maintenance and support requirements, the AI model must be able to be fine-tuned to improve its performance as researchers get more data. This can also help correct any issues with AI bias, which can hurt the animals. The fit criterion for this would be that researchers are able to replace the old AI model within only 1 hour of downtime. A similar down time should be expected for web application updates.

To make it easy for users, the look and feel of the application should be intuitive. In addition, there should be a welcome tutorial that displays the first time a user signs up, but can also be accessed later if users want to see it again. To ensure the stability of the system, the subsystems should always be connected and have minimal latency. For example, it should only take 1-10ms for the server to interact with the database. And in order to help the AI model, zoos must install video cameras with high resolution video in order to better see the animals and reduce the risk of errors. For legal requirements, the application must be certified by the Association of Zoos and Aquariums, and it must follow the Animal Welfare Act.

Design

The most important design goal of AnimalAware is the health and safety of the animals. In order to serve this, other design goals become important. For example, reliability is extremely important, because downtime could risk the lives of the animals. As discussed above, latency is also extremely important, and so are many other factors.

The overall system can be split into a few different sub-systems: the client, the server, the database, and the cameras. The cameras are simply physical hardware cameras that interface with the server and take commands such as when to go to sleep, when to power on and send data to the server, and more. The database subsystem stores the data for every model, and responds to queries and inputs from the server. Additionally, it also has a backup service, which has both local copies of the database and remote copies that can be loaded onto a cloud provider like Azure, to ensure the data is secure, even if the physical servers are destroyed by a ransomware attack or a natural disaster.

The server subsystem is where all the processing happens. It takes data from the cameras, correlates it with data from the database subsystem, and determines when to send alerts to the client. It also allows the client to update the database as needed, and can authenticate the users for such. The client subsystem provides an interface for zookeepers, managers, and vets to interact with the server. It also allows them to communicate with each other by updating records for animals and uploading their notes. Additionally, there are other features, such as to schedule vet checkups, for managers to approve them, for vets to receive the request and complete them, or to add or view information on the animals, or to add new cameras to the system.

Project Issues

AnimalAware is heavily reliant on live monitoring footage from the cameras installed around the zoo's animals. This could take a lot of money to set up, which could be a problem for non-profit zoos. Additionally, the cameras will have to be maintained, and could be damaged or destroyed by weather, natural disasters, or even the animals picking at them. On the bright side, AnimalAware's interface can be accessed through the browser, so it is compatible with almost all devices.

There are some current solutions that can help AnimalAware; however, none of them are designed to handle the production-level-scale that Animal Aware can. For example, one research project, Intellipig, is an attempt to monitor pigs' emotions using computer vision. This is a great start, and could be used to springboard AnimalAware's development into a production level system that can accurately and reliably manage the health and wellness of thousands of large animals. Additionally, there are existing database systems about animals. For example, The Zoological Information Management System (ZIMS) is the world's largest database of animals, used by over 1300 institutions. This data can be interfaced by AnimalAware.

Some possible risks that may interfere with AnimalAware are related to the centralization of data. Due to a high number of users across the globe accessing AnimalAware, and the systems and cameras uploading data to AnimalAware, performance becomes a key issue, especially considering the fact that the animals' lives are at risk. In order to mitigate the performance issue, AnimalAware will need to have expensive servers, which could drive the costs up. This could present a threat to the development of AnimalAware, especially if investors do not see it as a viable option. Additionally, because zoos are often non-profit organizations, they may not have the funds to pay for AnimalAware if it were expensive, so the price of the servers cannot simply be loaded onto the consumers.