**Project Specifications - SDZWA Sensor Team**

Ahmed Hussaini, Akshaya Sundaram, Brandon DalPorto, Ethan Tan, Jiping Lin, Malcolm McSwain

04/21/22

**Project Overview:**

The San Diego Zoo contains a vast network of devices to monitor various conditions related to the animals they take care of. While much of this network may be confined to a tight, interconnected architecture centered within the Zoo’s main perimeter, the Zoo also manages animals in more remote locations. These remote locations may contain a variety of sensors, all of which are collecting vital information such as weather conditions, sound, and video. Due to the nature of these sensors being geographically distant and difficult to reach, there arises a need to monitor the health of these sensors, and pull data from the source without a physical connection. With this problem in mind, we have been tasked by the San Diego Zoo to create a web application that provides a generalized and modular dashboard view of these sensor’s data. Our web application will provide support for various input formats, such as numerical, video, and audio - with capabilities of showing live feeds directly from the sensor. To complete this project, our team consists of a front end and back end team that will work together to integrate functional web apps with an administrative backend that allows for authentication and database management.

**Project Approach:**

Given the scope of CSE 145/237D, we have organized our team to work efficiently to meet required deadlines within the 10 week quarter. Our approach towards project management utilizes an AGILE workflow model, with a front-end and back-end team managing their respective milestones and deliverables - with the need for integration milestones as well. The front-end team’s work consists of web application development, centered around an interactive source for users to manage and monitor sensor data and health as it comes in from the Zoo’s provided infrastructure. The back-end team will be constructive database schemas for individual sensors, and an administrative back-end that manages data for authentication as well as sensor data. The front-end and back-end teams will be working closely together to ensure smooth integration between these projects, allowing for the front-end to display data based upon information retrieved from the back-end. In addition to inter-team cooperation, active communication between group members and Ian Ingram, our correspondent with the San Diego Zoo, will take place in order to meet specifications and expectations for the project’s scope. An example of this communication may be user tests for the front end UI/UX design to ensure everything looks and works according to their expectations.

**Minimum Viable Product (MVP):**

Our proposed MVP would provide the Zoo with a functional web app that allows for basic communication between the sensors and a modular web application that may retrieve specified information.

The MVP is as follows:

**Common Requirements:**

* System functional without internet connection
* The system will use HTTPS for communication with a self-signed or provided certificate.

**Front End Requirements**

* Dashboard capable of pulling data from multiple sources
* Dashboard web application can handle various data inputs:
  + Numeric
  + Video
  + Audio
* Displays within the dashboard include the following capabilities:
  + video/audio player
  + Line graph
  + Bar graph
  + Gauge graph
* Dashboard requires user authentication to access
* User may only access the dashboard if they received an invitation
* The project will use an already in place CRUD API

**Back End Requirements**

* Schema for sensor metadata
* Online Database for admin backend, which interacts with the frontend
* Server functionality for creation, deletion, and updating of admin database using REST API

**Constraints, Risk, and Feasibility:**

As with all projects, there are many risks and constraints involved which must be accounted for, as this plays a role in feasibility within the ten week scope of this project.

For constraints, the biggest concern is time and skill in the field, as our group must learn new tools over a short period of time and ensure proper functionality as is expected by the zoo. To mitigate this time constraint, careful planning has and will continue to take place regarding project management and milestones. An additional constraint for this project involves communication with Zoo administrators, as many of our group members have other classes and obligations, finding times to speak with employees at the Zoo may be hard to coordinate. A lack of communication in this project would lead to disaster between expectations and reality, as our goal as a team is to provide a viable product for the Zoo. In order to address this concern, we will provide regular updates via email to the Zoo administrators, as well as coordinate with Chris and Nathan times that work for conference call meetings.

Below is a table containing the various risks we face when working on this project as a group, especially with two separate teams:

| **Risk** | **Scale (1-5)** | **Assignment** | **Mitigation** |
| --- | --- | --- | --- |
| System Integration | 5 | All team members | Making sure that backend and frontend teams are corresponding closely to make sure that what each team builds is able to be easily integratable with the latter team's project and fits each other's needs. |
| Deployment in the Zoo Environment | 3 | All team members | Integrating our system with the Zoo infrastructure requires testing in a simulated environment as similar as possible. To do this, we will gather vital information from the zoo and attempt to obtain some of their devices / procure our own to ensure testing goes smoothly. |
| Retrieving information regarding sensors and infrastructure from the Zoo as well as a lack of familiarity with sensors. | 3 | All team members | Close discussion with the San Diego Zoo official over what sensors are going to be used and how information will be pipelined to our web application. |
| Unauthorized access/modification to database | 3 | Backend Team | Research and implementation of security practices for backend as well as a working authentication system which interacts with the frontend. |
| Front end is unable to process requests | 3 | Frontend Team  (Brandon) | Regular code review and testing would allow for us to catch bugs in the front end code quickly and ensure they may be fixed. |
| Unable to retrieve data for the front-end | 4 | Frontend Team  (Ahmed) | Work closely with the backend team and ensure that all required APIs may be implemented. Create well defined requirements to provide for the backend team. |
| UI/UX Expectations from the Zoo do not meet those implemented | 2 | Frontend Team  (Malcom) | Ensure constant communication occurs between Zoo and the team. Provide samples when applicable and retrieve user feedback. |

As discussions continue between team members, Chris & Nathan, and the Zoo, feasibility of the entire project will be determined. At this moment in the project development, we find that despite constraints and risks, the MVP as defined above is a feasible end product for the quarter.

**Group Management:**

Our team is currently split up into 2 subteams: one for frontend and another for backend, each consisting of 3 members. Decisions on frontend/backend specifications will be made by consensus by the specific team after consultation/check from the other team. Decisions that will affect overall project scope or integration will be made by the entire team by consensus. Communication within each smaller team will be done through public slack group chats, and overall team communication will be done using the sensor-dashboard slack channel. Sprints will be used for managing milestones, and ZenHub for backlog tracking. We will have one project facilitator that will manage our ZenHub as well as deadlines and team coordination.

**Project Development:**

The development team structure will also be based on frontend development and backend development team. For front end development, we will use ReactJS framework for the web interface and React Native if time permits for the reach goal of developing a mobile application. For the backend database, we will use SQLite for both administrative backend and for sensor metadata. The backend team will also try to use an Object-relational mapping (ORM) technique to build the administrative backend database. For web service and integration between frontend and backend, we will use NodeJS for backend and REST API wrappers for the web application. The deployment tool our team will use is Docker. For hardware related to this project, there will be various types of sensors according to San Diego Zoo’s requirements, and the data will be obtained from the sensors. At this time, we do not know what sensors will be in use other than an internet connected camera.

**Project Milestones & Schedule:**

**Frontend:**

Milestone 1: Deploy initial “hello world” web application

* + Our first milestone consists of deploying an initial “hello world” basic application which will allow us to familiarize ourselves with the ReactJS framework. Writing a program with simple webpage and button functionality is this first step.

Milestone 2: Registration/authentication flow

* + Our next milestone regards getting the workflow for the overall application up and working. The flow includes registration screen, authentication/login screen, and dashboard view. This milestone consists of initializing the project using ReactJS, creating the required views, and building out forms to capture user information. We want to create pages for each supported platform, and the dashboard will remain blank for now.

Milestone 3: Dashboard tile view

* + This milestone will be focused on getting the tile view of the dashboard up and running; this will be the primary front-end display component. We would also want a modular component to these tiles that can be added or removed, and give the user the ability to do this. There will be no graphing or data visualization in this dashboard yet.

Milestone 4: Graphing/data visualization functionality

* + In this milestone, we want to have graphing and data visualization functionality on the dashboard. This involves implementing the ability to produce a variety of different graphs and charts, depending on what forms of data need to be tracked. The graphs will be displayed on the “tiles”, and clicking on each tile will bring up a more detailed and interactive form of the graph.

Milestone 5: React Native application (stretch)

* + This final milestone would be to create a mobile port of the web application using React Native. The mobile app would have all the same functionality as the web application, just ported over to a mobile interface.

**Backend:**

Milestone 1: Set up a database schema for sensor metadata

* Identify a possible list of sensors that could be deployed in the zoo.
* Identify metadata sensor information for each of the sensors.
* Design an appropriate schema for the metadata per individual sensor.
* Map out the entity relationships manually or by using ORM (Object - relational Mapping)

**Implementation**:

* Implement the schemas using SQLite, and create tables to store metadata information per individual sensor.
* Completion output will be tables loaded with sample (or real data) containing sensor metadata information in SQLite database engine.

Milestone 2: Setup database schema & database for administrative backend

* Create a table to store user data and passwords for administrative frontend control.
* Create another table which holds data about sensors that users have access to look at.

**Implementation**:

* Host administrative backend online on a reliable Node server.
* Implement communication interface between the administrative backend and frontend.

Milestone 3: Writing server functions for additions/subtractions of sensors for the administrative backend database

* Create interactive functions like display and CRUD that the frontend user interface could use to query the backend admin interface.

**Implementation**:

* Generic API templates for implementing CRUD queries for all sensor tables.
* Work closely with the frontend team to write backend APIs for implementing frontend MVP features.

**Frontend Deliverable:**

The frontend MVP deliverable consists of a fully functional web application which uses ReactJS as the primary framework. This web app will allow invited users to authenticate or register a new account, upon successful authentication, they will be granted access to a modular, tile view of sensor data. This sensor data may consist of video, audio, or numerical information, as well as generic metadata which may indicate the health state of a given sensor real time, or in the past.

**Backend Deliverable:**The accumulation of the output of each of the aforementioned milestones will result in the backend MVP deliverable, which is a reliable backend support for storing sensor metadata, and administrative support. The deliverable will also support frontend administrative features, and will allow the admins to have CRUD access to the hosted database.

**MVP Schedule:**

All the milestones are given equal priority for the successful completion of the deliverable. The buffer time is given to accommodate any final changes to the milestone module after verification with the zoo. Testing, debugging, and integration will occur parallel to the implementation.

**Frontend Schedule:**

| **Week** | **Milestone** | **Team members working on Milestone** | **In - charge** | **Maximum Buffer Time - Manage risks** |
| --- | --- | --- | --- | --- |
| 5 | 1 | Frontend Team | - | 1 days |
| 5 | 2 | Frontend Team | Malcom | 3 days |
| 6 | 3 | Frontend Team | Ahmed | 5 days |
| 7 | 4 | Backend Team + Frontend Team | Brandon | 7 days |
| 8 | 5 | Frontend Team | - | 10 days |
| 9 | System Integration | Backend Team +  Frontend Team | Brandon | 3 days |
| 10 | Report, Video Recording | Backend Team + Frontend Team | Brandon / Akshaya | 1 day |
| 11 | Final Demo, Wrap up | Backend Team + Frontend Team | Brandon / Akshaya | - |

**Backend Schedule:**

| **Week** | **Milestone** | **Team members working on Milestone** | **In - charge** | **Maximum Buffer Time - Manage risks** |
| --- | --- | --- | --- | --- |
| 5 | 1 | Backend Team | Akshaya | 7 days |
| 5 | 2 | Backend Team | - | 3 days |
| 6 | 3 - Creating generic API templates | Backend Team + Frontend Team | Ethan | 3 days |
| 7 | 3 - Complete Integration with frontend | Backend Team + Frontend Team | Jiping | 5 days |
| 8 | API Templates for Collecting sensor data and feeding into database | Backend Team | Ethan | 3 days |
| 9 | API templates to check healthy conditions of some sensors | Backend Team | Jiping | 3 days |
| 9 | System Integration | Backend Team +  Frontend Team | Brandon | 5 Days |
| 10 | Report, Video Recording | Backend Team + Frontend Team | Brandon / Akshaya | 1 day |
| 11 | Final Demo, Wrap up | Backend Team + Frontend Team | Brandon / Akshaya | - |