Sensor Detection Applications with TIPPERS System

Author: Alpine Tang (*alpinet@uci.edu*),

Advisors: Professor Sharad Mehrotra, Dr. Roberto Yus

BACKGROUND

As the world around us becomes more interwoven with technology, the Internet of Things has made possible the idea of "smart" environments [1]. Through usage of sensors and monitors, data and information pertaining the respective environment can potentially be captured, which in turn can help developers optimize and improve all aspects within the environment (productivity, safety, energy usage, etc.). **TIPPERS** is a smart space system currently deployed at the University of California Irvine that utilizes sensors around campus to collect data about campus interactions. With this information, meaningful applications can be built and developed to efficiently reach a target group. For example, faculty can use occupancy data to make conscious decisions with regard to campus functionality, such as ensuring student safety by sanitizing commonly populated areas or minimizing energy usage in less commonly populated areas.

PROJECT GOAL

The goal of this project is to complete and improve applications by updating them to connect with the new TIPPERS backend system that returns more data and information than before. The applications I am working on on are Building Analytics, which reports information (occupancy, energy usage, temperature, etc.) in certain buildings or areas and the Self-Awareness App, which lets users monitor their own history of movement around campus.

REFERENCES

1. S. Mehrotra, A. Kobsa, and N. Venkatasubramanian. "TIPPERS: A Privacy Cognizant IoT Environment" in The First IEEE International Workshop on Security, Privacy and Trust for IoT, 2016.
2. P. Pappachan, M. Degeling, R. Yus, A. Das, S. Bhagavatula, W. Melicher, P. Naeini, S. Zhang, L. Bauer, A. Kobsa, S. Mehrotra, N. Sadeh, N. Venkatasubramanian. "Towards Privacy-Aware Smart Buildings: Capturing, Communicating, and Enforcing Privacy Policies and Preferences" in the IEEE 37th International Conference on Distributed Computing Systems Workshops

DESIGN

- With the new TIPPERS API, accessing data is much easier due to its simple user interface and functions
- Whenever an application requires sensor data, requests are made to the API, which in turn returns matching data from the server database
- The design of the old API was strict and limiting since each request could only access specified information. Figure 2 displays how different requests are needed solely for each room type being searched.

```
var API_INFRASTRUCTURE_GET = TIPPERS_HOST + "/infrastructure/get";
var API_CONFERENCEROOM_GET = TIPPERS_HOST + "/infrastructure/get?type=conference_room";
var API_SEMINARROOM_GET = TIPPERS_HOST + "/infrastructure/get?type=seminar_room";
var API_LOUNGESPACE_GET = TIPPERS_HOST + "/infrastructure/get?type=lounge";
```

Figure 2 . Old Endpoint Calls

- The new endpoint API returns data using less calls through the use of queries and method interaction. End calls with user parameters let requests access a larger variety of data and reduces the redundancy of similar requests.


```
let baseUrl = process.env.REACT_APP_BASE_URL;

export default {
  entity: baseUrl + "/entity",
  observation: baseUrl + "/observation/" + process.env.REACT_APP_OBSERVATION_ID,
  query: baseUrl + "/test/query",
}
```

Figure 3, 4. New Endpoint Calls

- Access to SQL queries further eases data requests by using connections between certain data types to facilitate necessary information. Figure 5 demonstrates how queries enable quick searches through large amounts of data.

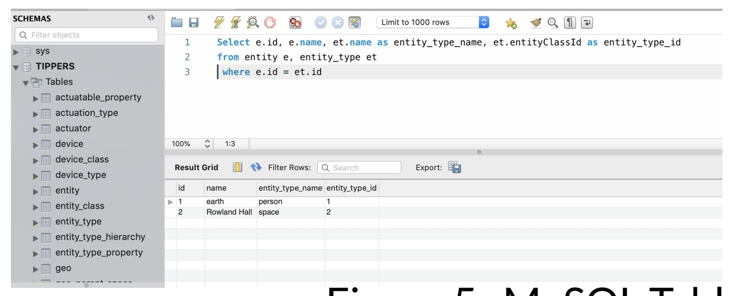

Figure 5. MySQL Table

FUTURE PLANS

With the new API implemented, its functionality can be used to develop more applications that track other sensor data. We plan to implement applications that track energy usage (how electric/heat can be optimized in areas depending on occupancy), temperature (which areas require more energy depending on its location), and sensor health (monitors activity of sensors). [2]

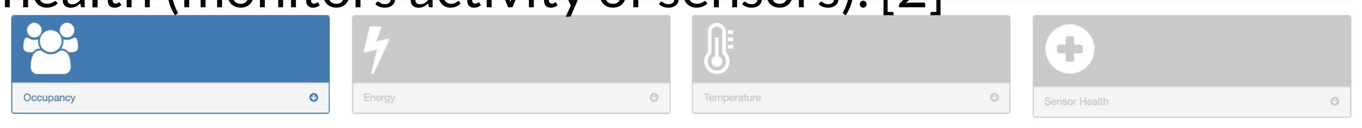

Figure 6. Future Applications

APPLICATION

- **Building Analytics** is an application that creates a visual display of occupancy within certain areas through a specific timeframe. Faculty and others use the graph to compare trends between population in areas, and these trends help facilitate planning/design of applications within campus.
  1) The left hand side displays all the types of areas on campus (i.e: buildings, spaces, rooms, etc.). Users select from there which areas they want the data of.
  2) Above the graph, users can select what time frame they are looking for, and the interval for which the data points will be plotted.
  3) A respective graph is created to display each occupancy data point, and connecting lines are automatically formed to generate a trend for the occupancy in the area. Figure 5 shows a graph of occupancy in Donald Bren Hall and Rowland Hall between the time period of 06/27/2020 and 07/01/2020.

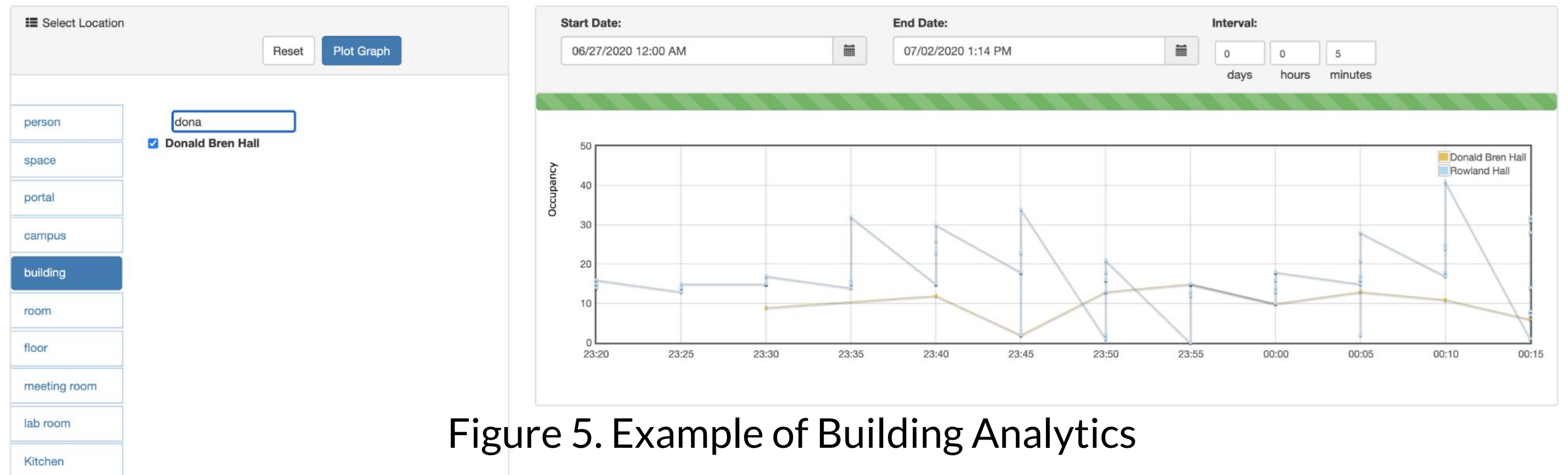

Figure 5. Example of Building Analytics

- **Self-Awareness App** acts in a similar manner, but it uses sensor data to track personal movement and location rather than buildings and spaces. Sensors around data keep track of which users are at each locations, and by compiling it all together, a user can see where they've been, who they've interacted with, how many steps they've taken, etc. Figure 6 demonstrates the statistics a user can see about themselves.

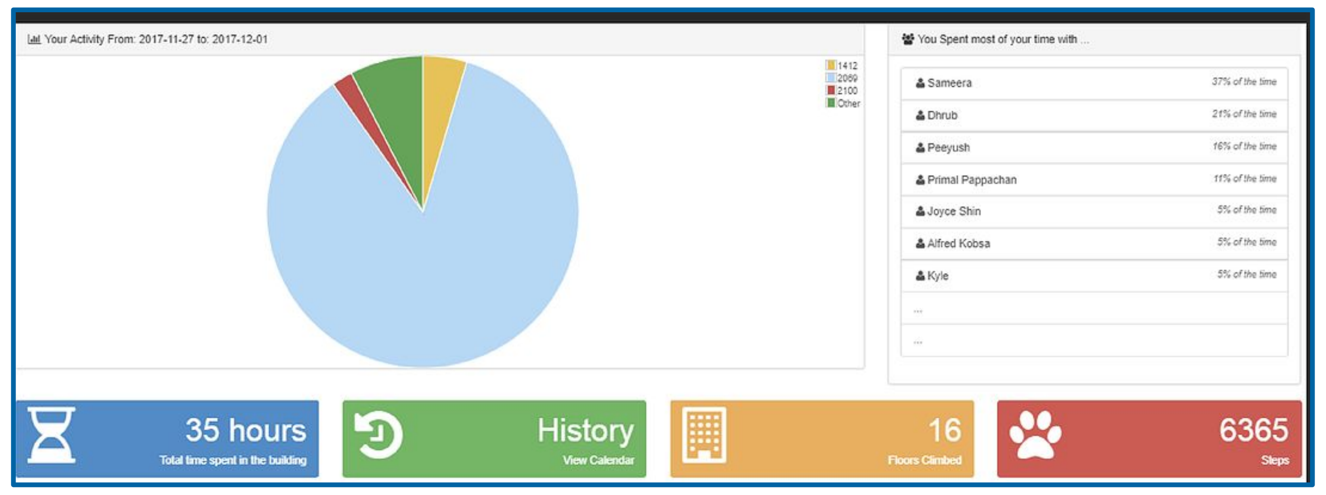

Figure 6. Example of Self-Awareness App