Test Report – First Prototype

To: Professor Pisano

Team: 7 (MuseumMate)

Date: 11/19/2023

Group Members:

John Culley jculley@bu.edu

Kai Imery kimery@bu.edu

Yanyyang Zhang yz21@bu.edu

Ananth Sanjay ananths@bu.edu

Kwadwo Osafo kosafo@bu.edu

1.0 Equipment and Setup Summary

The equipment used in the prototype we presented can be split between two broad categories; hardware and software. The software consisted of separate applications which run the server, frontend, handheld user devices (TourTags) and beacons. The backend consisted of various code which describes the functions of our networking and storage capabilities. Our hardware consisted solely of Adafruit Huzzah32 ESP32 Feather Boards, which were split to serve as either a TourTag (housed in a 3D enclosure) or a beacon, as well as our personal computers which served as physical servers for the software components. Suitable power supply units were also attached to each of the ESP32 boards.

The physical setup of our prototype consisted of plugging the beacons into electric sockets at various predetermined locations in PHO 111 and PHO 113. These predetermined locations are shown as **[B#]** in **Figure 1.** The TourTag was also connected to a mobile rechargeable power unit and placed in a 3D printed enclosure. Next, the backend (Node.js server) and the frontend (React Native application) were started on two of our computers. The log output from our respective applications allowed us to verify whether each part of the system was integrated properly at startup. After confirming successful connections, we then proceeded to demonstrate the communication and location identification aspects of our system. After that demonstration was completed, we connected an RC522 RFID module to one of our TourTags to demonstrate the functionality of the RFID aspect of our system. Below is a detailed list of all the materials used for our setup:

Hardware:

- **Adafruit Huzzah32 ESP32 Feather Board**
- **3D Printed Enclosure for Handheld Devices (TourTag Enclosure)**
- **5V Power Supply**
- **PKCELL LP552535 3.7V 420mAh**
- **RC522 RFID Scanner Module**

Software:

Server:

- **Node.js**
- **Express.js**
- **InfluxDB**

Frontend:

- **React Native**
- **Expo**

User Device:

- **BLE Scanner**
- **RFID Reader**
- **Connect to Campus WiFI**
- **UDP Client**

Beacon:

- **BLE Advertiser**

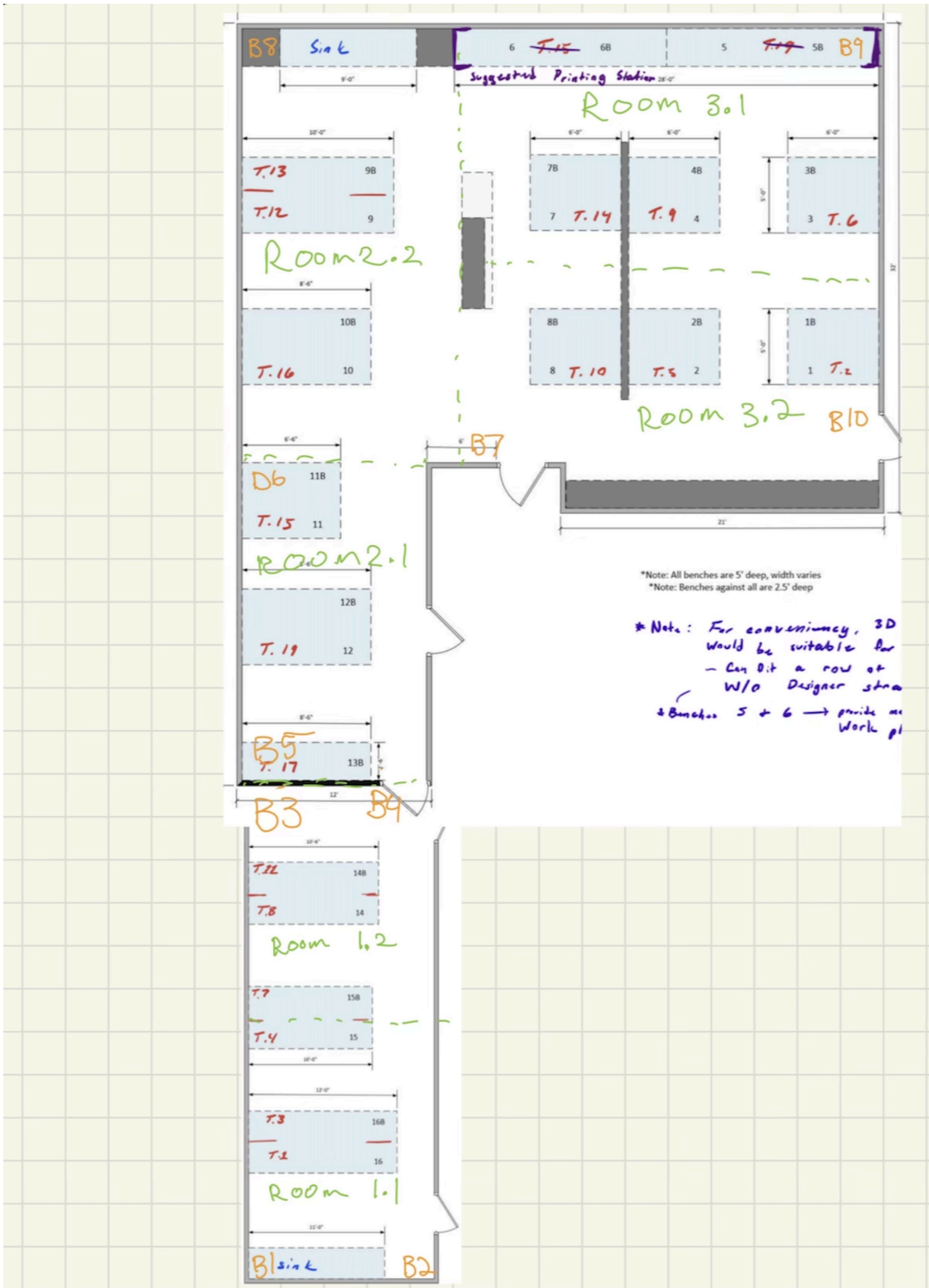

Figure 1: Locations of beacons and room sections

2.0 Testing and Measurement Summary

Included in our test plan was a list of measurable criteria that defined vital components of our system and was proof of proper functionality of our prototype. Below is the list of proposed measurable criteria. It is important to note that references to the number corresponding to each measurable criteria in the list will be made in the sections it precedes.

1. ESP32 Beacon advertises BLE messages continuously
2. ESP32 User Device collects beacon BLE strengths every 3 seconds
3. ESP32 User Device connect to eduroam via WPA2-PEAP
4. ESP32 User Device sends {beaconID, userID, signalStrength} to the Node.js server
5. ESP32 User Device reads RFID Tags
6. Node.js server updates signal values in the data structure
7. Node.js returns a location from HTTP endpoint given a UserID
8. The app should initialize and display the HomeScreen within 2 seconds after launch, providing users with the main navigation options.
9. The BarcodeScanner function must activate the device's camera, successfully scan a barcode, and display the result on the screen within 5 seconds of the user's command.
10. When selecting the TourTypes option, the app should display a list of available tours, including but not limited to TimedTour and Explore tours, within 3 seconds.
11. For the CurrentLocation feature, the app must access the HTTP endpoint, accurately determine the user device's current location, and retrieve the relevant map image to display on the screen within 6 seconds of user engagement.
12. Throughout the navigation of the app, the transition between screens should be smooth, with no screen freeze longer than 2 seconds.
13. All interactive elements on the screens such as buttons, and clickable links should respond to user input with no more than a 2-second delay.

Our testing procedure began with setting up and starting the hardware elements and the backend and frontend. Next, the application was rendered on a mobile phone in order to demonstrate the user interface to the system. The mobile application allowed the user to use the phone's camera to scan a QR code on the handheld device (TourTag) enclosure, which in turn allowed the application to identify what TourTag the user possesses, and this information is used to query the backend for the specific location information of the TourTag. The functionality of various buttons and screens on the mobile application was also demonstrated. The application was then navigated to a screen which displayed a map similar to that in Figure 1 which was configured to serve as a live map with an indicator showing what room the user is currently in.

The next step in our demonstration involved moving the TourTag to different rooms in order to demonstrate the functionality of the live map feature. Our TourTag obtained RSSI values of the strength of BLE signals between itself and the beacons, which continuously broadcast the signals. Every 3 seconds, the signal strengths measured were sent to a server on our backend over WiFi through UDP in the form {beaconID, userID, signalStrength}. The data received was stored in an InfluxDB database through the use of an API. Our server then used a pre-trained KNN machine learning model to predict the location of the user based on the signal strengths and outputs this location. The frontend is then able to query the backend through an HTTP endpoint for the location information. The frontend was configured to allow for strict adherence to the measurable criteria in order to eliminate any latency and successfully meet the expectations.

The last step in our demonstration consisted of connecting an RC522 RFID module to one of our TourTags to demonstrate the functionality of the RFID aspect of our system. The TourTag was connected to a computer which produced specific logs whenever the RC522 module was touched by an RFID tag.

2.0 Conclusions

Overall, our demonstration was a success and we were able to meet all the proposed measurable criteria. We received useful feedback from the reviewers of our prototype. The reviews consisted of suggestions regarding shifting away from machine learning as a means of predicting location, and instead focusing on simply relying on RSSI values. Whereas this suggestion will reduce the complexity of our project in terms of both software and hardware required, we ultimately conclude that machine learning predictions will be pivotal in our project. This is because we observed how RSSI values can fluctuate depending on the amount of traffic and architecture of the room. Using machine learning will allow us to overcome this challenge and will make our system more robust and versatile. Furthermore, we determined that using machine learning does not increase the complexity of our system by any substantial metric. Consequently, we plan to increase the amount of data being fed as training data to our machine learning model, as well as experimenting with other types of machine learning algorithms to deduce which model might best suit our needs.