First Prototype Test Plan – 11/16/2023

To: Professor Pisano

Team: 7 (MuseumMate)

Date: 11/16/2023

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 Required Materials

Hardware:

- **Adafruit Huzzah32 ESP32 Feather Board**
- **3D Printed Enclosure for Handheld Devices**
- **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**

2.0 Prototype Architecture

The prototype we are demonstrating can be split between two broad categories; hardware and software. The software consists of separate applications which run the server, frontend, handheld user devices (TourTags) and beacons. The backend consists of various code which describes the functions of our networking and storage capabilities. Our handheld devices (TourTags) send data to a server on our backend through UDP and data received is stored in an InfluxDB database through the use of an API. The frontend consists of a React.js application which allows users to log into our system and receive positioning data corresponding to their respective TourTags. The last part of our software consists of code which allows the TourTags to record BLE signal strengths between themselves and the beacons, as well as to send this data to the server through the use of WiFi.

Aside from our personal computers which currently house the software to run the backend and frontend elements of the system, our hardware consists solely of Adafruit Huzzah32 ESP32 Feather Boards, which are split to serve as either a TourTag (housed in a 3D enclosure) or a beacon. The beacons are powered by 5V power supply units connected directly to electrical sockets, and the TourTags are powered by PKCELL LP552535, which is a rechargeable mobile power supply unit. RFID modules will also be attached to the TourTags to demonstrate its functionality.

3.0 Setup Procedure

Pre-testing Setup Procedure:

Physical Setup:

1) Plug in Beacons into wall sockets using 5V power supply units in designated locations shown in **Figure 1** ensuring that the ESP is facing the right way to allow the most precise positioning
2) Plug PKCELL power supply unit into User Device

Software Setup:

1) Run the command **node index.js** to start backend
2) Run command **npx expo start** to start frontend

4.0 Testing Procedure

1) Render app
2) Run Node.js Server
3) Scan QR code on TourTag enclosure
4) Follow Screens Navigation
5) Display Current Location of User Device from HTTP query

5.0 Measurable Criteria

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.

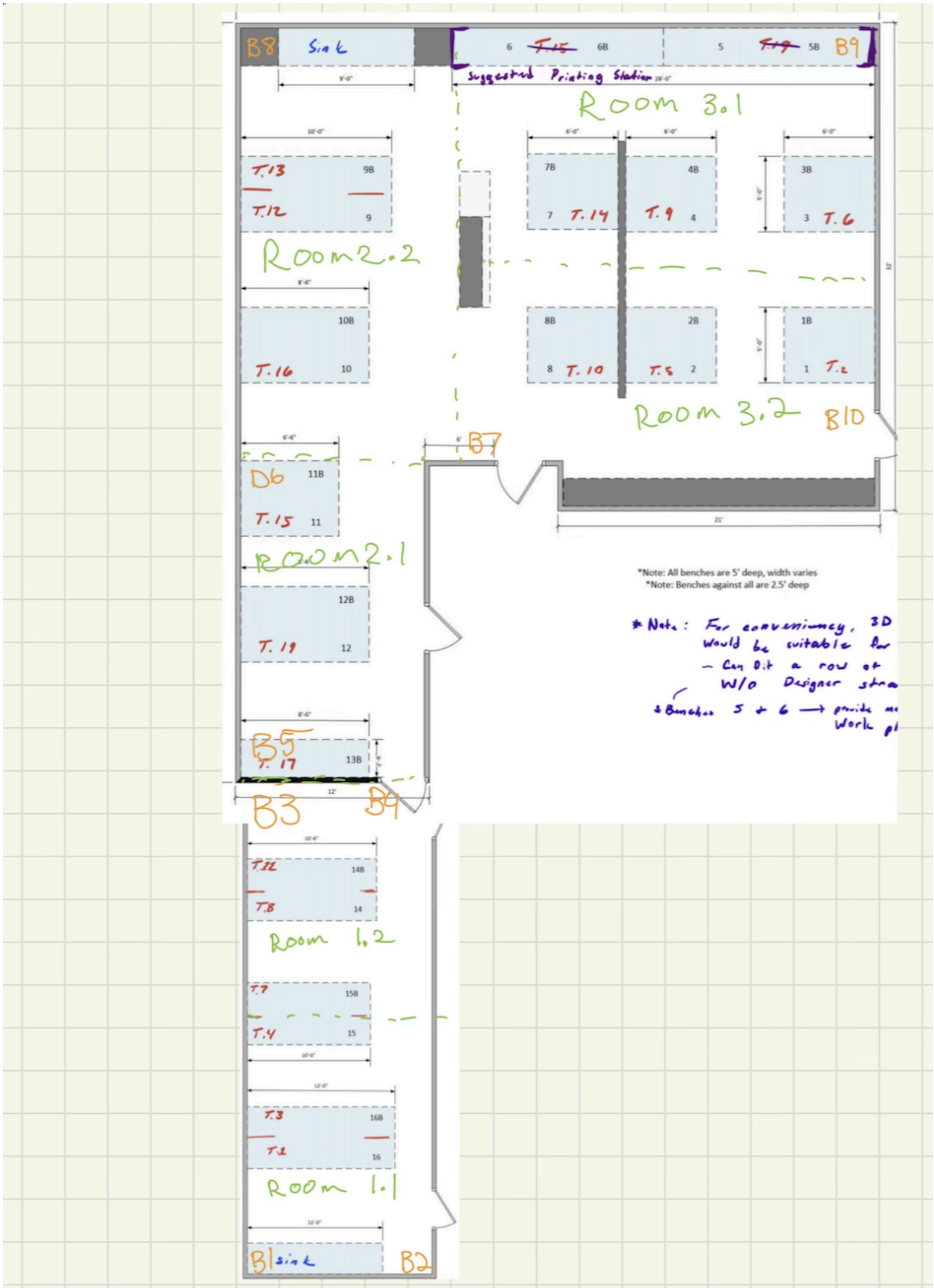

Figure 1: Locations of beacons and room sections