Test Report – Final Prototype

To: Professor Pisano

Team: 7 (MuseumMate)

Date: 04/11/2024

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

In the culmination of our efforts leading up to ECE Day, our project, MuseumMate, was showcased on the 9th floor to illustrate the full spectrum of its functionalities, focusing on enhancing the museum visiting experience through technology. This demonstration was a critical part of our final testing phase, highlighting the seamless integration between hardware and software components. The hardware improvements were particularly notable with the assembly of Printed Circuit Boards (PCBs), providing the backbone for the operational efficiency of our devices, including both the TourTags and beacons. The server setup marked a significant milestone, hosted within the building on a local machine in our lab utilizing Dynamic DNS (DDNS). This ensured a stable and reliable network address, pivotal for the seamless communication and processing of data. The setup on the 9th floor was meticulously arranged, with UWB beacons positioned to ensure optimal coverage and accuracy in location tracking, facilitating an immersive and interactive demonstration of the system's capabilities in real-time.

Central to our system's infrastructure is the local server, set up within the lab environment, serving as the nerve center for our operation. This server, running on a dedicated machine, hosts our backend applications and manages data flow to and from the MuseumMate devices. Utilizing Dynamic DNS (DDNS) technology, the server maintains a consistent internet presence, ensuring that data from the TourTags can be transmitted efficiently, regardless of any changes in the network's external IP address. The server is responsible for a multitude of critical functions, including processing the location data received from the TourTags, managing the database where this information is stored, and serving as the backbone for our admin dashboard. By hosting the server locally, we ensured that data transmission was both fast and secure, key considerations in the context of a live museum environment where real-time tracking and data integrity are paramount.

On the software front, the introduction of an admin dashboard stood out as a highlight, offering museum administrators an intuitive interface to access and manage the data collected throughout the museum. This connection to InfluxDB underscores our commitment to providing a tool that not only enhances operational efficiency but also offers insights into visitor interactions and behaviors. The deployment also featured our advanced user interface, accessible through React.js applications, designed to enrich the user experience with detailed positioning data, multimedia overlays, language options, and text-to-speech functionalities, thus paving the way for an interactive and educational visit.

Hardware:

- **Makerfabs ESP32 UWB**
- **Adafruit Li-Ion/Li-Poly Charger [v1.2]**
- **3D Printed Enclosure for Beacons**
- **PCB for Handheld Device**
- **USB 5V Power Supply**
- **PKCELL LP552535 3.7V 420mAh**
- **Passive RFID Tags**
- **RC522 RFID Scanner Module**

Software:

Server:
- **Node.js**
- **Express.js**
- **InfluxDB**
- **MinIO**

ChatGPT Server:
- **Gin Framework**
- **ChatGPT API (3.5 Turbo)**
- **Golang**

Frontend-App:
- **React Native**
- **Expo**
- **React**

Admin Dashboard:
- **React.js**
- **AntDesign**
- **InfluxDB**

User Device:
- **UWB Module**
- **Battery Charging System**
- **RFID Reader**
- **Connection to Campus WiFi**
- **UDP Client**

Beacon:
- **UWB Module**

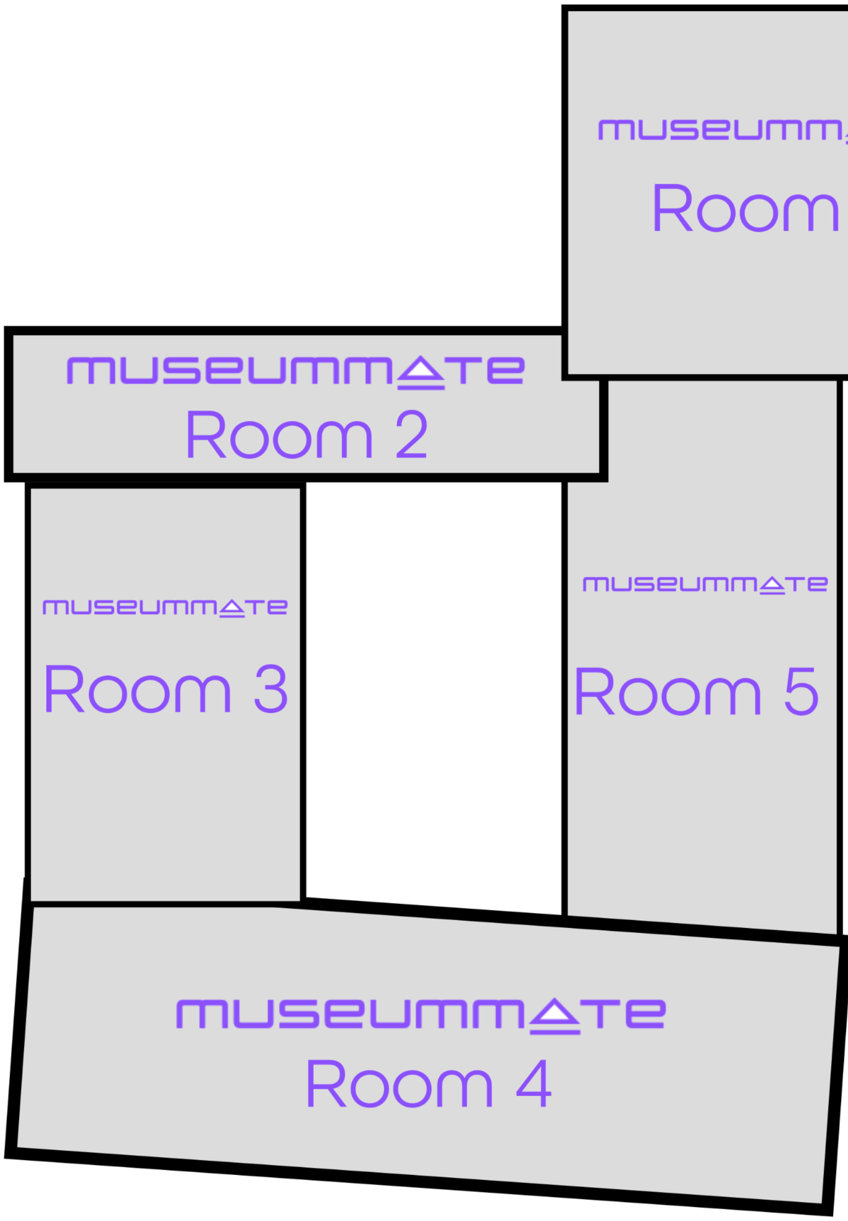

Figure 1: Locations room sections on the 9th floor

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.

Hardware:

1. Handheld (User) devices accurately deduce the distance between themselves and a given beacon within approximately +/- 0.2m error bounds.
2. 3D printed enclosures should not interfere with the WiFi and UWB antennas of the enclosed beacon
3. RFID tag is successfully scanned by handheld device, and the corresponding RFID tag ID is recorded.
4. Updated distance measurements from UWB readings should be sent to the central server upon the detection of signals from 3 unique beacons in a single sampling interval.
5. Contents of packets sent via UDP to the central server should be consistent with predetermined protocols.

Server:

1. Successfully receives UWB data via UDP from user device
2. Stores and processes UWB data with a queue that is updated each time new data is received
3. API Testing:
    a. /location/:userID (GET)
        i. Returns user location to the front end
    b. /path/:userID (GET)
        i. Given a userID and room, return route to get to destination
    c. /tsp-path (POST)
        i. Given an array of rooms, return the shortest path to visit all the rooms
    d. /rfid/:bucketName (GET)
        i. Given a bucket name, return all associated multimedia from minIO
4. In 5-second intervals, occupancy by room is obtained and sent to InfluxDB to be timestamped

ChatGPT Server:

1. API Testing:
    a. /chat (POST)
        i. Returns the response from ChatGPT based on the prompt given in the request
        ii. Upon successful request, response is either retrieved from cache or if it is a new prompt it is generated with the ChatGPT API and returned

Front-End-App:

1. Initializes the app and swiftly navigates to the HomeScreen within 2 seconds, offering primary navigation.
2. Activates the BarcodeScanner to utilize the device's camera for scanning a barcode and displaying results within 5 seconds.
3. Displays a list of available tours, including TimedTour and Explore, within 3 seconds upon selecting TourTypes.
4. Accurately locates the user's device and retrieves the corresponding map image within 6 seconds using the CurrentLocation feature.
5. Ensures smooth screen transitions without freezes exceeding 2 seconds and responsive interactive elements within 2 seconds.
6. API Testing:

a. /location/:userID (GET)
i. Retrieves the current location of the user's device and updates the display or UI elements.
b. /path/:userID (GET)
i. Provides a navigation route for a specified userID to a selected room within the app.
c. /tsp-path (POST)
i. Receives a user's selected rooms and computes the shortest visitation path.
d. /rfid/:bucketName (GET
i. Returns all multimedia associated with a given bucket name from minIO.
e. WebSocket ws://128.197.53.112:8080
i. Establishes real-time connection for RFID data, updating UI with new object info.
f. Google Translate API https://translation.googleapis.com/language/translate/v2
i. Translates text to selected user language for improved accessibility.
g. GoLang API http://128.197.53.112:4040/chat (POST)
i. Accepts prompts and returns informative responses for interactive user engagement.
h. /api/exhibit-rating (POST)
i. Accepts the rating of an exhibit and posts it to the server which inturn stores the rating in InfluxDB.

Admin Dashboard:

1. Rendering Performance:
    a. Dashboard Load Time: Ensure the admin dashboard's initial load time does not exceed 3 seconds, from login screen to full dashboard visibility.
    b. Data Visualization Rendering: Test all data visualizations (graphs, charts, maps) to render within 2 seconds of data request initiation.
    c. Live Data Feed: Live data feeds, including visitor counts and real-time location tracking, should update on the dashboard without manual refresh and within a 1-second delay from data transmission.
2. Navigation and Responsiveness:
    a. Screen Transition Speed: Screen transitions within the dashboard (e.g., switching between data views or accessing settings) should occur within 1 second, ensuring swift navigation
    b. Interactive Elements Response Time: Interactive dashboard elements, such as dropdown menus, sliders, and buttons, should respond to user interaction within 500 milliseconds.
    c. Scrolling and Zoom Performance: Scrolling through data lists and zooming in on maps or charts should be smooth, with no visible stutter or delay.

3. User Interface (UI) Consistency and Accessibility:
    a. UI Element Consistency: All UI elements (buttons, icons, text fields) should maintain visual and functional consistency across different screens of the admin dashboard.
    b. Accessibility Features: Test accessibility features such as text size adjustment, color contrast settings, and keyboard navigation to ensure they meet accessibility standards and provide a seamless experience for all users.

Our demonstration sequence initiated the operational setup of both hardware components and the software backend. Following this, we showcased the administrative dashboard to highlight its comprehensive monitoring and notification capabilities. Subsequently, we transitioned to the mobile application, initiating with the user interface presentation. Demonstrating the application’s functionality, we employed the mobile phone's camera to scan a QR code attached to the TourTag device. This process enabled the app to identify the specific TourTag in possession, utilizing this data to fetch precise location information from the backend server.

We proceeded by navigating through the app's various screens, ultimately arriving at a feature that allows users to select a custom tour. For this demonstration, the tour commenced in room 4, proceeding to room 5. As we moved, the app dynamically updated to guide to the next step, illustrating the seamless integration between the user's physical movement within the museum and the digital guidance provided by the app. Additionally, we highlighted the dashboard's capability to broadcast notifications to all users in real-time via WebSocket connections, enhancing the interactive experience.

Further demonstrating the app's multifaceted functionalities, we explored the translation services, text-to-speech features, and the integration of ChatGPT for enriched informational content. These features were showcased alongside the app's rating system, offering a holistic view of the user engagement and interaction possibilities within the museum environment.

Concluding our demonstration, we integrated an RC522 RFID module with one of our TourTags to exhibit the RFID technology in action. By tapping the TourTag against an RFID chip, we triggered multimedia content to be displayed on the mobile app, successfully testing the app's ability to deliver content in multiple languages, utilize text-to-speech functionalities, and provide interactive ChatGPT-based information. This comprehensive testing phase not only validated the functional integration of the system components but also highlighted the enhanced visitor experience facilitated by our application.

2.0 Conclusions

The feedback and outcomes from this final testing phase have been instrumental in refining our system. A key takeaway was the need for a balanced approach in updating the location of each TourTag to enhance the user experience while conservatively managing device battery life. The introduction of dynamic polling, with modifiable rates based on user activity, represents a significant advancement in this direction, optimizing the balance between real-time tracking accuracy and battery conservation. As we approach ECE Day, MuseumMate stands as a pioneering solution, merging technology with cultural education to transform the museum experience. Our system not only promises an unparalleled interactive and educational visit for museum-goers but also establishes a new benchmark for interactive learning environments. The journey to this point has underscored the potential of innovative technology to enrich cultural and educational experiences, setting the stage for MuseumMate to inspire and educate in equal measure.