Deliverable H

Group 12: Brayden Baker, Shailen Mann, Nicholas Pighin, Jaron Roy

GNG1103

1. Abstract

This document defines the 3rd zone restriction prototype from group 12. It details the steps taken to combine the separate subsystems into one functional, comprehensive prototype. It details the multiple tests performed on this prototype and their results. Feedback was once again collected from potential clients and was used to improve the final prototype. The bill of materials has once again been updated with the coding for this prototype being transferred over to JavaScript in place of Python.

2. Defining the Prototype

Our third prototype combined the separate subsystems into one comprehensive and functional prototype.

3. Iteration on Previous Prototypes

In the previous prototyping stages, our results indicated that the math and formulae used to determine the UE's distance from the nearest boundary, as well as the check to determine whether the UE is within the boundary, were accurate. However, they took over 200 lines of code to execute, and were written in Python, which is somewhat difficult to integrate with HTML, and which is notoriously slow. As such, this prototype saw the complete rewriting of these algorithms, now in JavaScript, to enable a speedy and easy integration with the GUI subsystem.

The GUI itself was also further developed, as, while it was proven to work as expected during prototyping stage 2, it was not very engaging to look at. As such, this prototyping stage involved adding CSS elements in order to make it look more like how we would like our final product to resemble.

Twilio's text messaging API, previously described, was also integrated into our website, though doing so required much more work than expected, as a local server must be running on the user's computer in order to be able to send text messages. The software for this server was developed during this prototyping stage with the help of generative AI, and was simplified in order to allow for ease of use for future clients.

Finally, though the Excel documentation system was proven to work, for the most part, during the first prototyping stage, it became clear during this prototyping stage that potential users did not see themselves interacting with this potential feature much, so it was scrapped from this comprehensive prototype to enable the dedication of more time to more essential subsystems.

4. Prototype III Test Plan

4.1. Test 1

Test Descriptions	Verifies whether the prototype correctly detects when a user crosses a boundary triggers an appropriate alert in real-time
Reason for Prototype	Ensure Safety and Operational Efficiency --- Identifying breaches and addressing them promptly
Evaluation Criteria	Alerts are triggered upon boundary crossing --- Alert contains accurate information
Level of Prototype	High fidelity --- Focused
Kind of Prototype	Visual
Metrics	Proportion (%) --- Delay (s)
Analysis Method	Log occurrences and times of triggered alerts for review
Stopping Criterion	When the UE completes 10 crosses without any unexpected system behaviour
Results	The system successfully detected all 10 crossings and displayed accurate alerts in real time. Alerts were correctly dismissed when the clear button was clicked. No false positives or missed detections were observed.
Interpretation	Pass

4.2. Test 2

Test Descriptions	Evaluates how different components of the zone restriction system work together to provide a seamless experience.
Reason for Prototype	Testing integration of multiple functions --- Individual components communicate effectively, preventing system failures
Evaluation Criteria	Alerts are generated when boundaries are crossed --- All components respond within acceptable time limits
Level of Prototype	High fidelity --- Focused
Kind of Prototype	Visual
Metrics	Proportion (%) --- Delay (s)
Analysis Method	Log system timestamps for each process (location update, alert generation, zone modification) and calculate response times.
Stopping Criterion	The test stops after successful verification of component interactions under two different scenarios: 1. No boundary crossings (system remains silent). 2. Boundary crossing detected and alert generated.
Results	The system demonstrated effective integration across all components. Location tracking maintained an accuracy of ±1 meter, alerts were generated in under 1 second. No conflicts or lags were observed, confirming seamless communication between components.
Interpretation	Pass

5. Feedback Collection

From Potential Users

- “It would be helpful if a text was also sent when the user has entered back into the restricted zone”

- Reorganize page so that alerts take up less space and mini-map is much bigger and in the center of the screen

- Give buttons a hover-over description

- Visibly change color of buttons after they have been pressed

- Add a compass to mini-map

- Add option to display detailed alerts or just whether UE is within bounds

- Logging all alerts in Excel is not a priority

6. BOM Update

Component	Quantity	Unit	Unit Cost	Total Cost	Link
Visual Studio Code	4	EA	0.00	0.00	https://code.visualstudio.com/
Shabodi Workshop	1	EA	0.00	0.00	
Python 3.13	4	EA	0.00	0.00	https://www.python.org/downloads/
Microsoft Excel	1	EA	0.00	0.00	
Twilio API	2	EA	0.00	0.00	https://www.twilio.com/console/projects/summary?_gl=1*1hql8zh*_gcl_au*NDI5MzMyMTg1LjE3MzEwMDI5NTg.*_ga*MTI1Nzc0MTczNy4xNzMxMDAyOTU4*_ga_RRP

					8K4M4F3*MTczMTExMDE2Ny4zLjEuMTczMTExMDIzNC4wLjAuMA..
ECMAScript 2024 (JavaScript)	4	EA	0.00	0.00	https://nodejs.org/en/download/package-manager
HTML5	4	EA	0.00	0.00	https://html.spec.whatwg.org/multipage/
CSS3	4	EA	0.00	0.00	https://www.w3.org/TR/CSS/#css
				Total	
			Added Total	0.00	
			Final	0.00	

7. Conclusion

After the completion of prototype 3 we were able to successfully implement our separate subsystems defined in the past prototypes into one comprehensive and functional prototype. In the past prototypes our code was split between Python and JavaScript. To combine these separate prototypes more simply all the code was changed to JavaScript. An entire new subsystem was also added at this stage with that being a system to interpret data and generate alerts to be sent out. The GUI was also improved to make it look more appealing. This prototype was focused on the functionality of each of these subsystems integrated together and to see the general functionality. With the systems now working together effectively the team is feeling confident about the creation of the final product.