Deliverable D

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

GNG1103

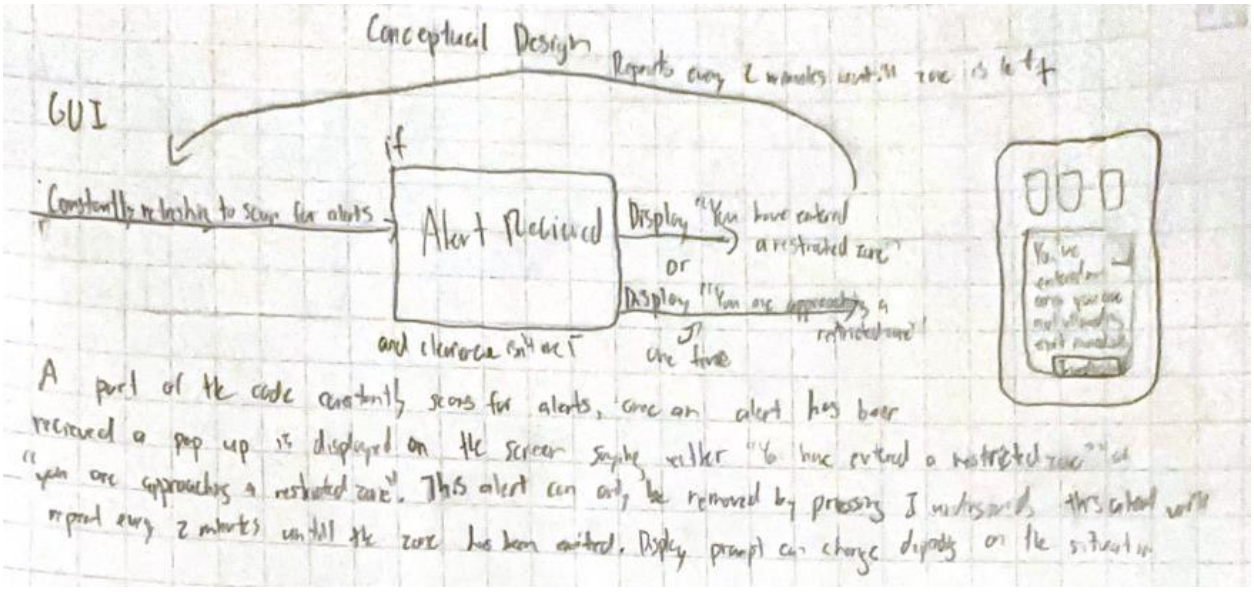
1. Abstract

This document represents our ideas for each subsystem that we use to choose the best idea for each subsystem. We considered the feasibility of each idea and how good the design idea was compared with the criteria that must be met to make these decisions. The first subsystem is the GUI (graphical user interface) which mainly takes into account how the system will be accessed and controlled as well as general aesthetics. The second subsystem is location tracking which means how we will go about monitoring and tracking the UEs such as a scanner searching for signals from UEs or GPS. The third subsystem is alerts which describes how and under what conditions alerts will be sent out and who those receiving the alerts will be. The fourth subsystem is zone definition describing how we will go about defining the zones we need to restrict and separating them, this will be a subsection of the GUI. The fifth subsystem is connection meaning how the components of the system will connect with each other and transmit information. The document lists a series of subsystem designs by each group member and then compiles the best ones determined by the group into a table that weighs the various aspects of each one. The report finalizes by generating a series of solutions based on the data above and identifies the most optimal global concept.

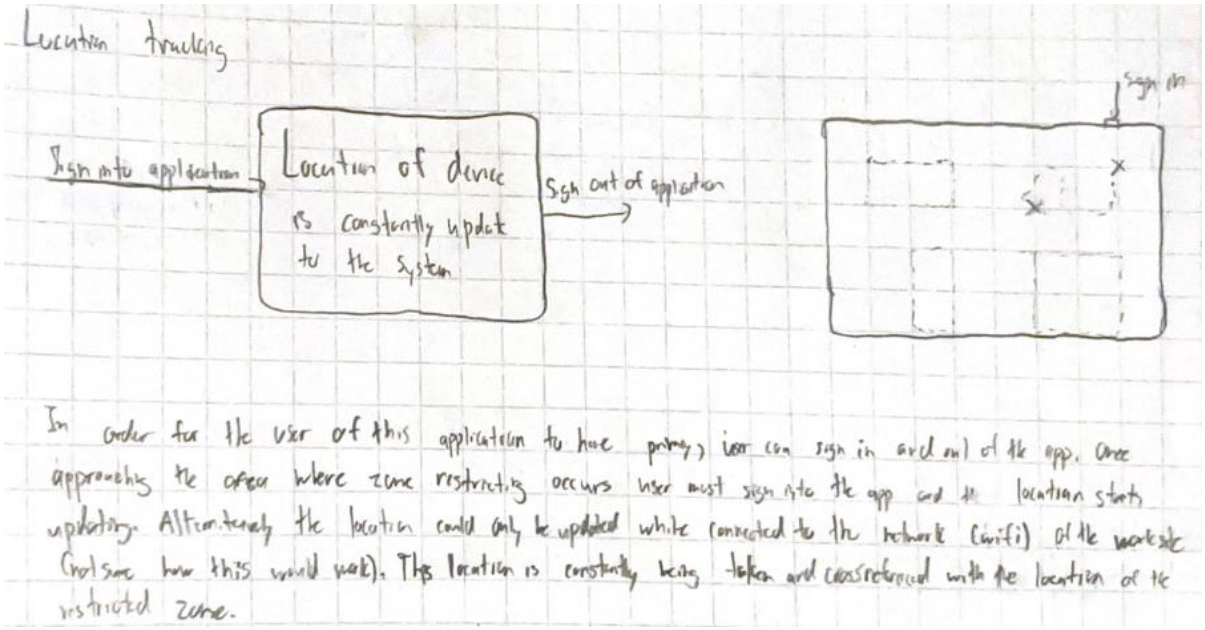
2. Original Subsystem Concepts

2.1. Brayden

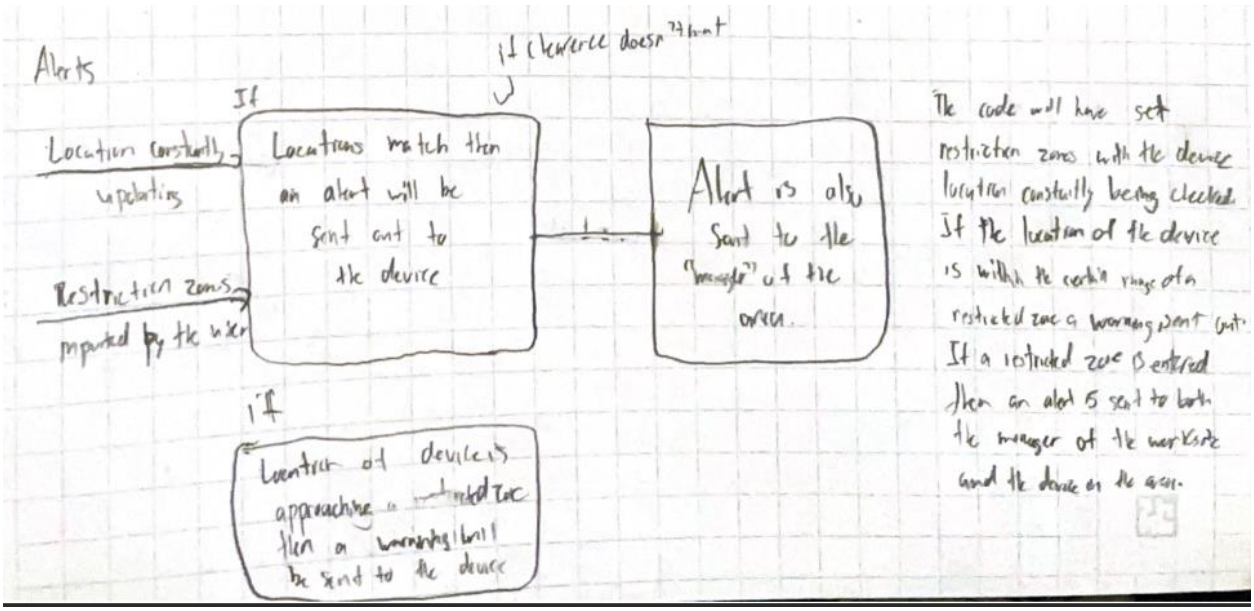
2.1.1. GUI



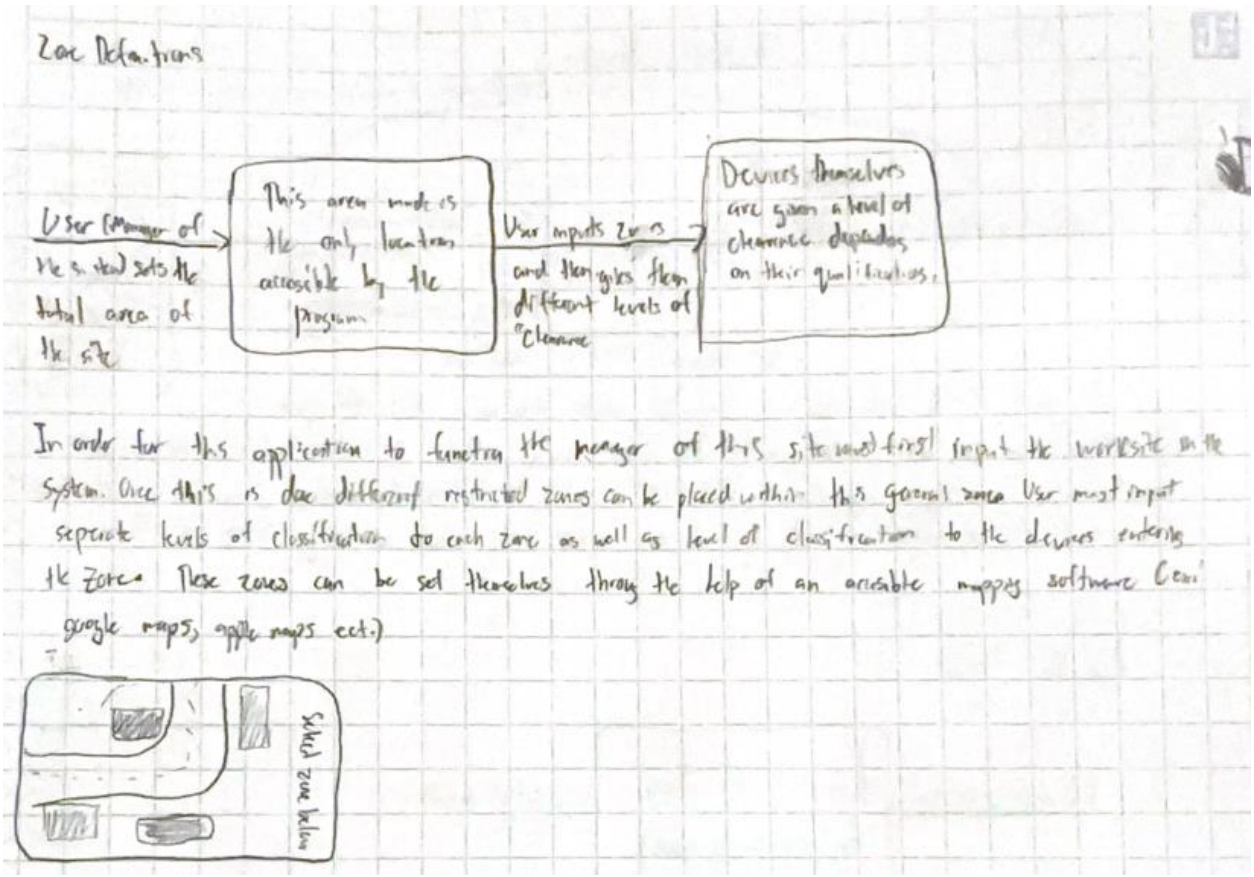
2.1.2. Location Tracking



2.1.3. Alerts



2.1.4. Zone Definition



2.2. Shailen

2.2.1. GUI

- Home screen will show all zones with every UE and where they are located

- Can click on each individual zone to “zoom in” and have a clearer view of what is taking place in the zone along with nametags for all UE’s

- Can pin up to a certain number of UE’s to see view them on the home screen

- Will also be a tab to see a list of each UE and its current location

- Can click on the UE and be brough to the zone map with a focus on that certain UE

- Will be able to change what type of UE’s are allowed in certain zones, the speed of certain zones, the total cap of UE’s in the zone

2.2.2. Location Tracking

- Dots on the map will show their current location

- Can have a general signal sent out to record the position of every UE at a specified time interval (2 seconds, 5 minutes)

- Have the setting modable for every single UE (some get updates realtime/faster than others)

2.2.3. Alerts

- Maybe only work with real time tracking

- Machine Specific: Alert when coming close to a machine for any personnel. There will also be an audible beep/alarm from the machine

- Machines should not be able to cross the restricted zone however in the event they do an alarm will be triggered from the machine to alert all nearby personnel

- Alert will be triggered when near the edge of the particular zone and a different one when crossing it

- When entering a new zone have the UE either ask for permission to enter or just upload its new location

2.2.4. Zone Definition

- Using a relay in some area it is possible to create an origin

- The zones will then be created by a series of points/lines on the plane

- The more points created the better the zone definition

- For machines some zones may be assigned different speeds

- Zones can also have a cap on how many UE's are allowed in at once and what type are allowed in

2.2.5. Connection

- Connected through radio waves

- Wifi

2.3. Nick

2.3.1. GUI

A purely text-based GUI (excluding zone definition), which prompts users to enter specific numbers to open menus, exit features, etc. Whenever an input requires more than just a number to specify which action to take, a clearly identifiable text input field will be displayed to the user. An example is shown below:

You are at the main menu. Please input a number specified within brackets to open its corresponding sub-menu.

[1] Alerts

[2] Zone Definition

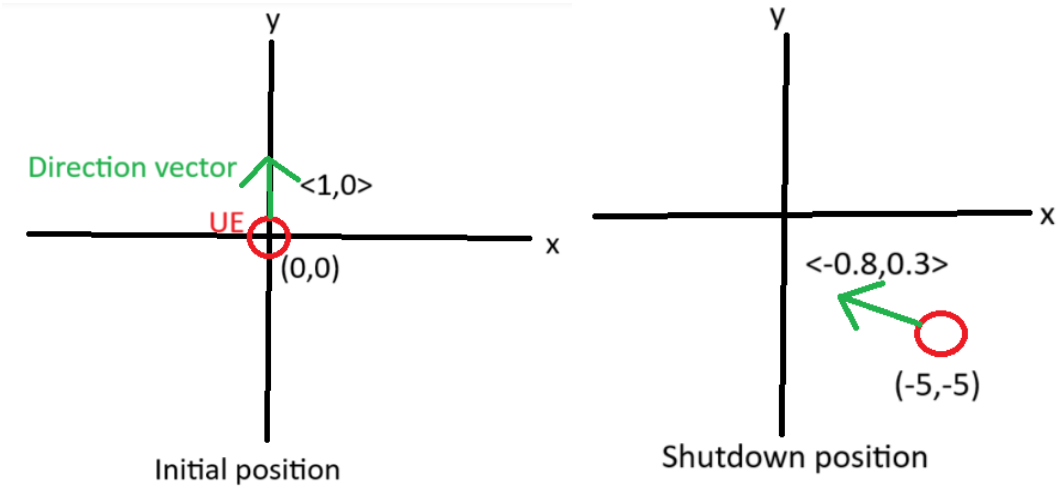
[3] Connection

[4] Exit

2.3.2. Location Tracking

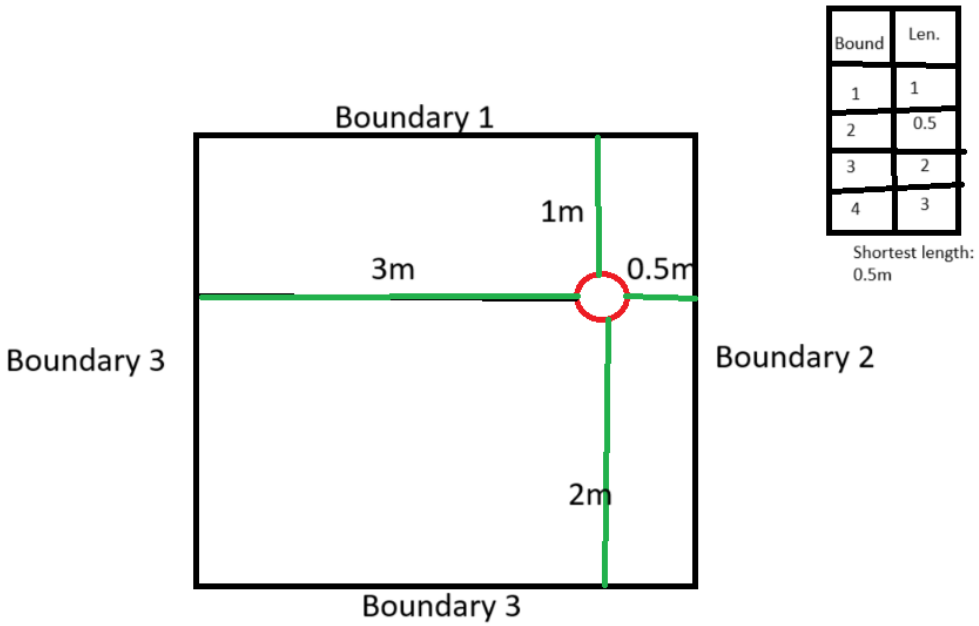
2.3.2.1. Concept 1

The initial location of the UE will be stored as (0,0), with the coordinates representing 1m on a Cartesian plane. Its initial direction vector will be stored as <1,0>, or directly along the +y axis. As the UE moves, its location will be updated using its velocity in a direction multiplied by the time it travels in that direction. Whenever the UE is shut down, it will send its location, direction vector, and ID to a database, where both will be stored until the UE regains power, at which point it will have its initial location and direction vectors set to the values they were most recently stored as.



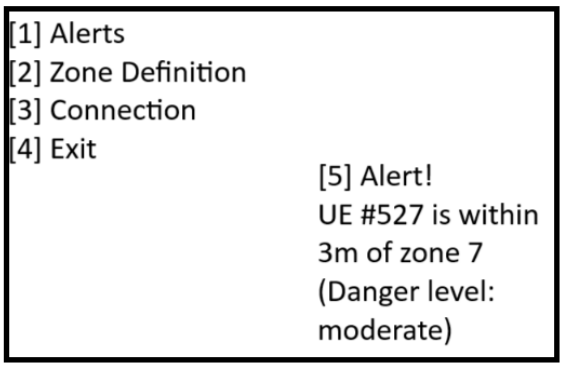
2.3.2.2. Concept 2

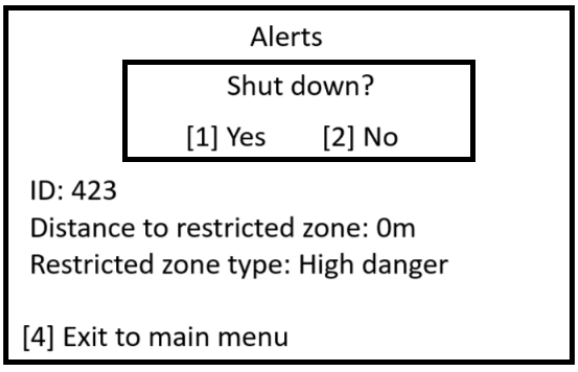
The direction vector of the UE will be tracked, as in Concept 1, but the location of the UE will instead be stored as an array, whose values are the distances of the UE from each edge of its boundary (calculated as the length normal to the boundary edge between the UE and the boundary). This array's values will initially be set to measured distances from the setup location of the UE to its boundaries, and they will be updated by the method described in Idea 1. The program will determine the lowest value in the array, and will report this as the proximity of the UE to its boundary.



2.3.3. Alerts

Whenever a UE is within a user-specified distance to a boundary, a notification will appear on the GUI with all relevant information (UE ID, distance from boundary, zone danger level, etc.). There will be an option to open this notification to take further action, as well as one to dismiss it. If a UE crosses a boundary, the user will be instantly redirected to the alerts menu, and will be prompted with an option to stop the UE remotely. Along with this option, they will also be given all relevant information to the alert.

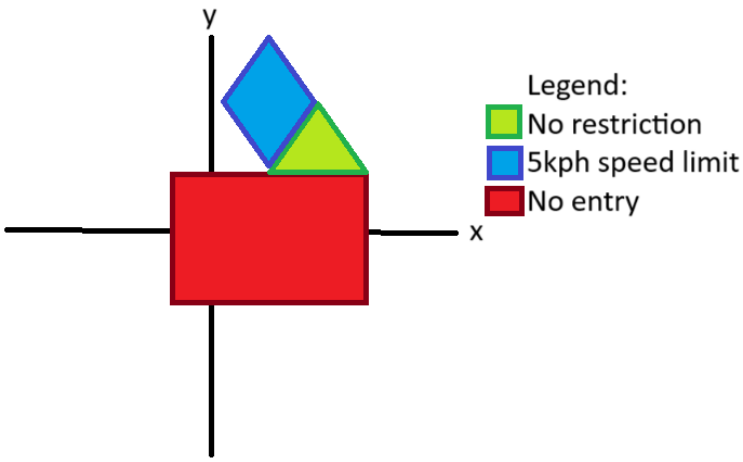




2.3.4. Zone Definition

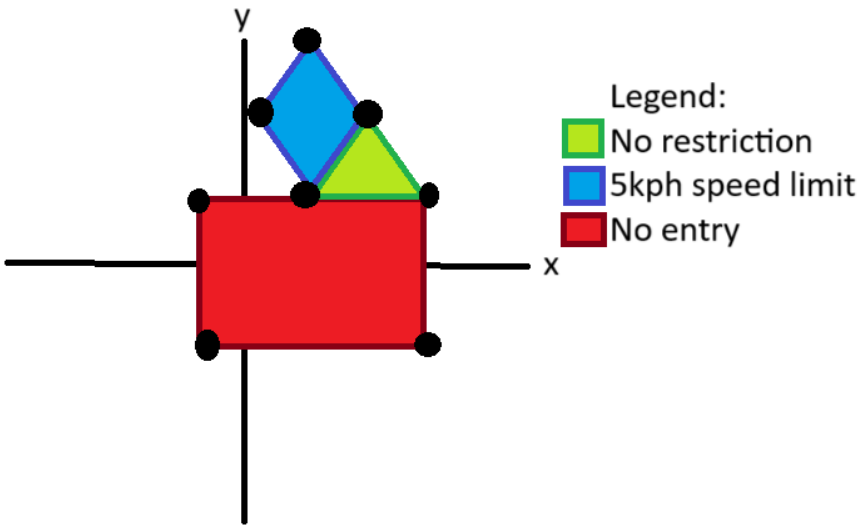
2.3.4.1. Concept 1

In a graphical interface, the user will be able to place lines on a Cartesian plane to create polygons, and will then be prompted to specify the UE's restriction level within these polygons (no restriction included) with user specified properties/actions for different levels (i.e. should the UE instantly stop, can it only enter if there are <5 other UEs in the area, what should it do within a certain proximity, etc.). In this prompt, they will also be able to select a "New Zone Type" option, in which they can create a new danger level. The positions of the vertices of these polygons will be stored in a table of arrays, which will include the location danger level corresponding to each polygon-defined restricted zone.



2.3.4.2. Concept 2

All will remain the same from Concept 1, but instead of the user placing lines on a Cartesian plane, endpoints which can be used to create those lines will be auto-populated by corresponding location markers in the real world.



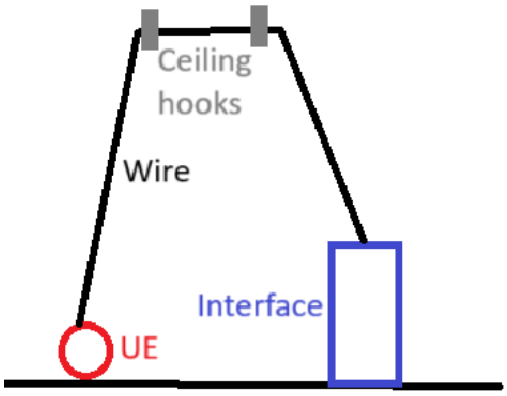
2.3.5. Connection

2.3.5.1. Concept 1

Using Bluetooth, UEs will be connected to an interface. This Bluetooth connection will need to allow both read and write, as the interface will need to get and interpret location, etc. data from the UEs, and will also need the ability to stop, or otherwise modify their behavior, remotely.

2.3.5.2. Concept 2

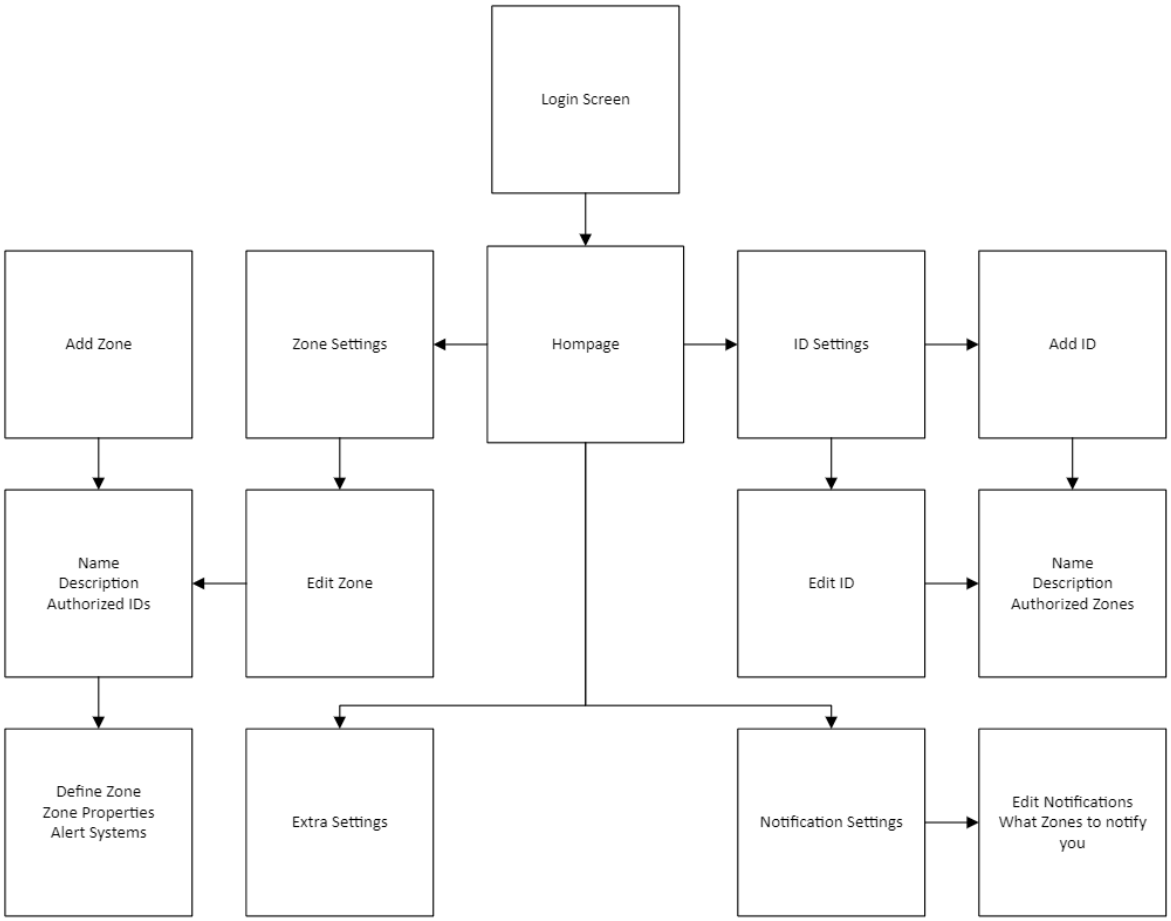
The UEs and interface will be connected by physical wires which run through the ceiling, which will, as in Idea 1, need to allow both read and write. This wire will be retractable to avoid the UEs running it over/other complications.



2.4. Jaron

2.4.1. GUI

The base design is an interactive GUI that allows users to select what path they wish to follow and add information to it. Different sections allow for managers to give descriptions of zones and id and organize different properties by name.



2.4.2. Location Tracking

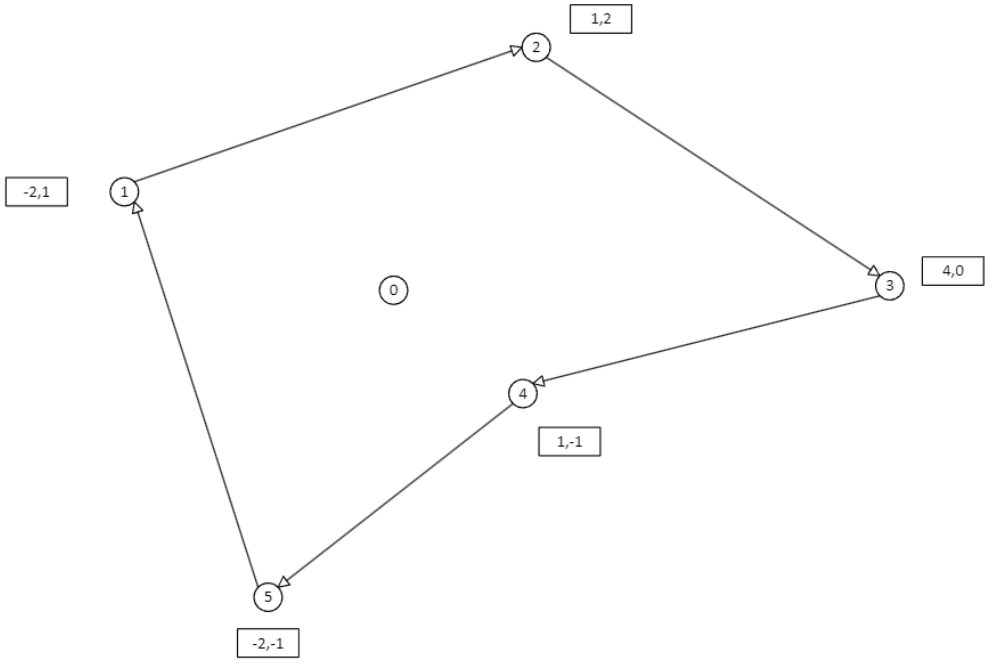
This tracking system revolves around a central scanner instead of a GPS system, like how radar looks. The scanner would scan radially, either with a rotating scanner or a complete 360 reader. This scanner would search for active RFID signals, which when found would determine the distance from the scanner and the angle of rotation to plot its position.

2.4.3. Alerts

Having the application connect and notify users can hard, especially if they dont allow notifications. Therefore, in the event of an individual entering a restricted zone, alarms and lights can be connected to the system and set off notifying the individual that they are not supposed to be there. In addition, those responsible for a zone can be contacted as the system can expect them to react and want to receive notifications.

2.4.4. Zone Definition

The zone can be defined graphically by using a cartesian plotting system. With GPS or other trackers, define a central landmark that the sytem can be calibrated around then define corner points relative to the origin. Each corner point will connect in numerical order before returning to the first point.



2.4.5. Connection

The different physical components of the system can connect over private networks and make calls to APIs to function. This will also help ensure security as only individuals with network access can easier access the data being transmitted.

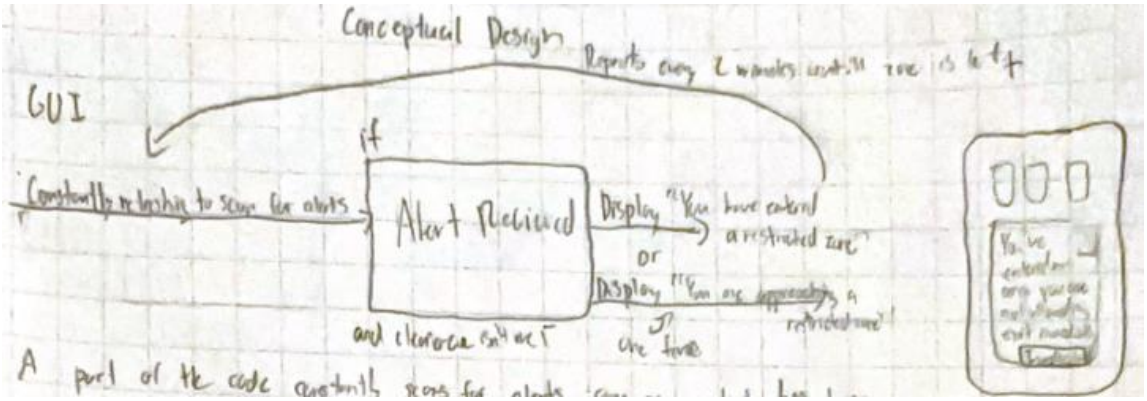
3. Modified Subsystem Concepts

3.1. GUI

3.1.1. Alert-Centered GUI

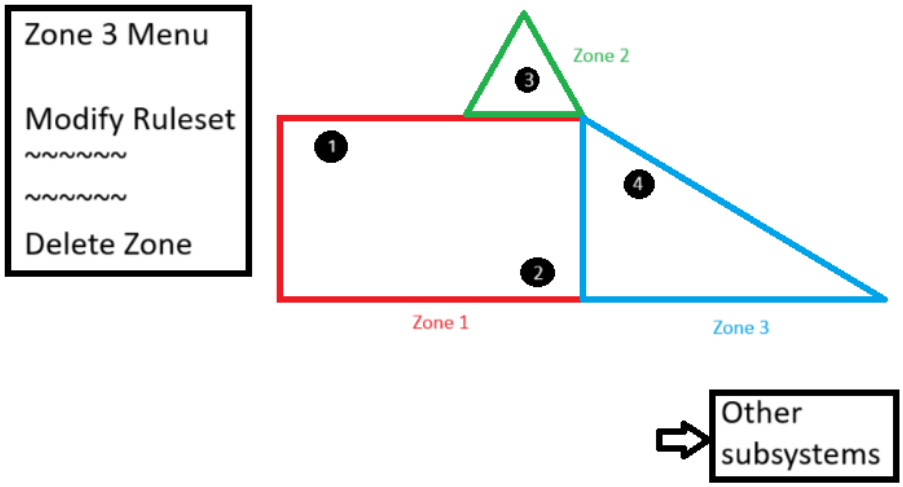
As in Brayden's GUI subsystem idea, the home screen, as well as the general functionality, of the GUI will be focused on any incoming alerts for UEs which are approaching or entering restricted zones. Adding on to it, there will also be options to access the other subsystems of the

project, but these will take a backseat to the important problem of timely alerts. This concept would likely greatly increase safety in the workplace, as there would be a heavy emphasis on preventing accidents by knowing as soon as possible about errant UEs before they can cause any harm. However, accessing any of the functionalities of subsystems other than Alerts would be a lot more difficult, and the development of those other subsystems would also likely be stunted, as having such a GUI would likely make us focus much more heavily on Alerts than anything else.



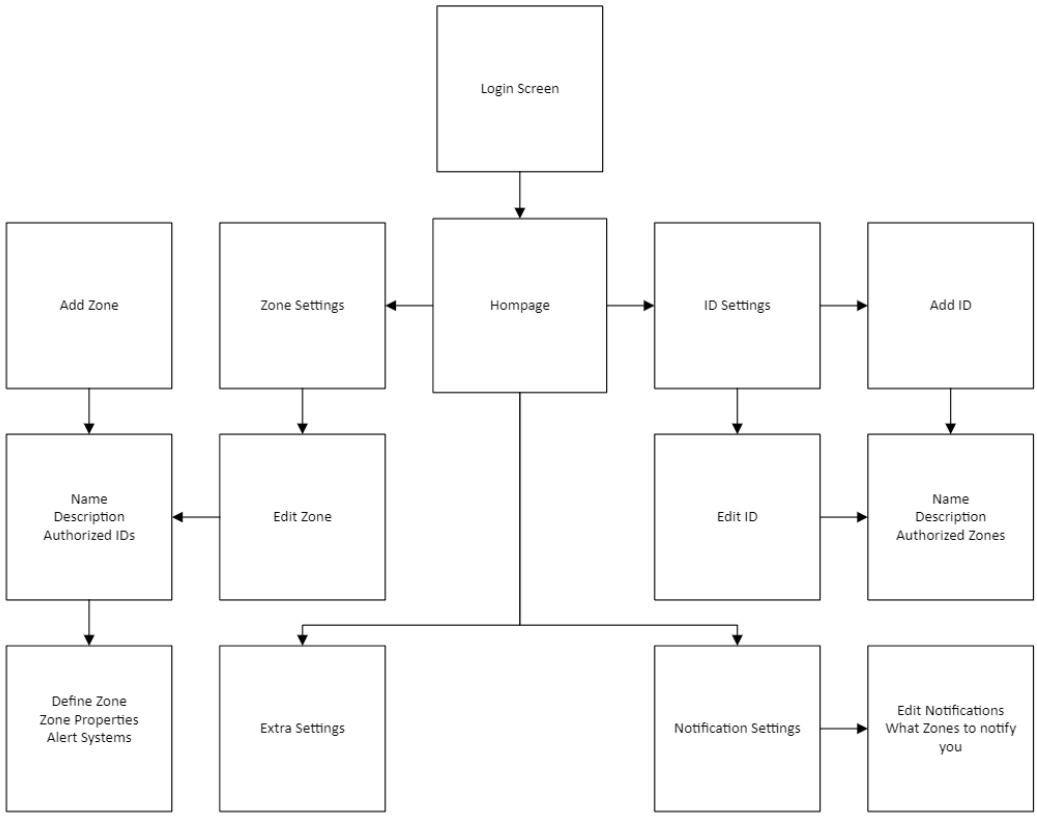
3.1.2. UE Position-Centered GUI

As in Shailen's GUI subsystem idea, the GUI home screen will display the position of all UEs within the restricted zones defined by the user. Moreover, this modified idea would also involve the abilities to modify UE information by clicking on them; change, add, or remove zones with a simple menu; and go to the Alerts and Connection subsystems by clicking on a menu off to the side. Similarly to alert-centered GUI, this concept would be great at one specific function: understanding where UEs are at any given moment, and would be very useful in an environment where there are frequent changes to UE and zone information. However, it would struggle in terms of allowing users to quickly stop UE, as the alerts menu would be more difficult to access, and the development of both Alerts and Connection would likely be stunted.



3.1.3. General GUI

Instead of focusing on specific subsystems with the GUI, this concept involves a much broader home screen, as with Nick and Jaron's ideas. From the home screen, the user would have easy access to any of the other specified subsystems and their functionalities. While this would make the product, as a whole, easier to navigate, as there would be no 'hidden' systems which would take more effort to find than others, the home screen would not serve much of a function at a glance, as it would with the other 2 subsystem ideas. Moreover, all other subsystems would likely get an equal amount of development time, which would make for a more cohesive product in terms of completeness, but which might result in no truly complete subsystem with our given timeframe, where the other, more specific GUI concepts would likely result in at least 1 or 2 other subsystems getting completed.



3.2. Location Tracking

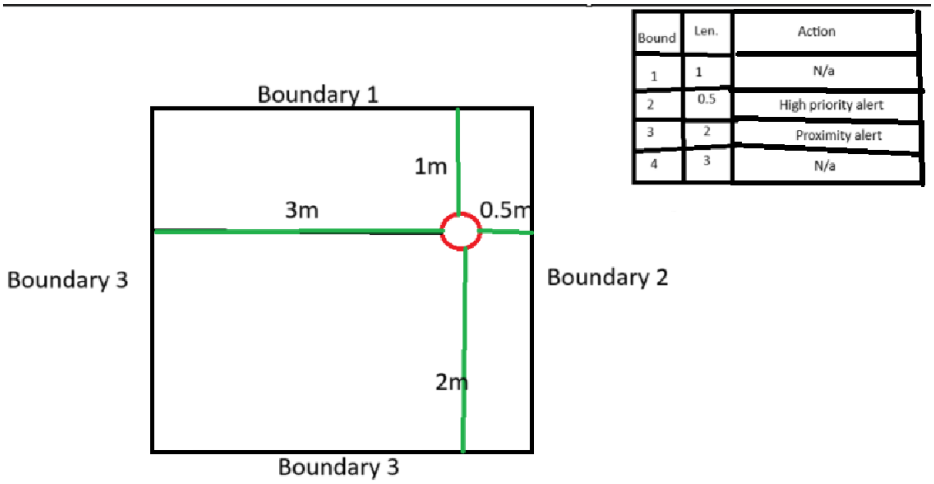
3.2.1. Variable Time Tracking

As not all UEs necessarily bring the same amount of risk to the workspace, this location tracking concept offers the user the option to change the time interval between alerts from each specific UE, like Shailen's concept, as well the information categories (ID, distance from specific zones, etc.) in each update. In terms of actually tracking UE locations, this concept is not well defined, which could lead to struggles when it comes to determining development milestones. However, it would be a very useful tool in a workspace where there are many different types and categories of tools, as it would allow for a user to prioritize alerts coming from the UEs they deem most important.

UE #432	6m from zone 1, 2m from zone 2	1min ago
UE #52		2min ago
UE #432	5m from zone 1, 3m from zone 2	2min ago
UE #432	10m from zone 1, 1m from zone 2	3min ago
UE #15	3 nearby UEs	3min ago
UE #432	1m from zone 1, 5m from zone 2	4min ago
UE #432	3m from zone 1, 3m from zone 2	5min ago
UE #52		5min ago

3.2.2. Boundary Tracking

As the UEs only need to be restricted from entering certain areas, their positions mostly only matter in terms of how far they are from where they are not allowed to go. Building from Nick's 2nd concept for this subsystem, then, the location of each UE will be represented as its distance from each boundary, instead of its true position in space. However, instead of only caring about the smallest distance, as in the original subsystem concept, this modified concept will consider all distances, and will base its alerts and other actions based on the ruleset of each boundary it is near. While this concept would allow the user to easily visualize the position of each UE, as well as their distance from specific danger zones, it would not be very adaptable to preventing UE collisions with each other, for example, and would likely involve a lot of data transfer if there were a lot of UEs and/or restricted zone boundaries.

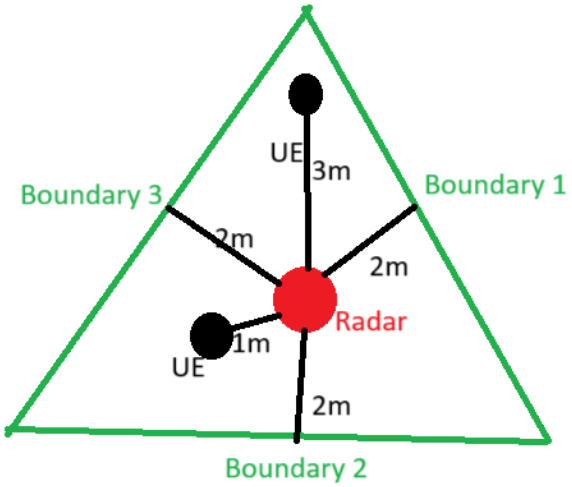


Bound	Len.	Action
1	1	N/a
2	0.5	High priority alert
3	2	Proximity alert
4	3	N/a

3.2.3. Radar Tracking

This modified concept is based off Jaron's original Location Tracking concept. Since tracking the position of an object on the scale of global coordinates, or any other large area, can be quite

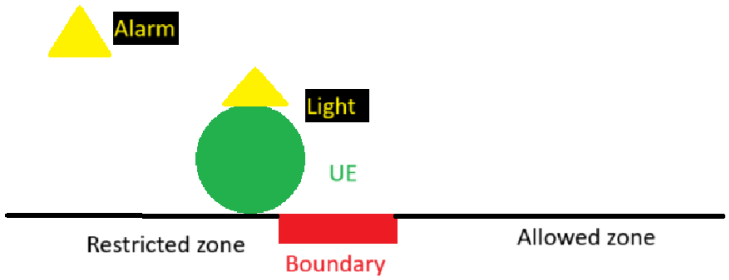
inaccurate when it comes to the precision needed for restricted zone devices, a radar tracker in the middle of the workspace would be very useful for getting exact distances of both UEs and boundaries. However, there would be a much larger initial setup cost than other solutions, and might run into problems if there are physical boundaries between the radar and the things it is supposed to track, as well as having trouble if a boundary is hidden behind a UE.



3.3. Alerts

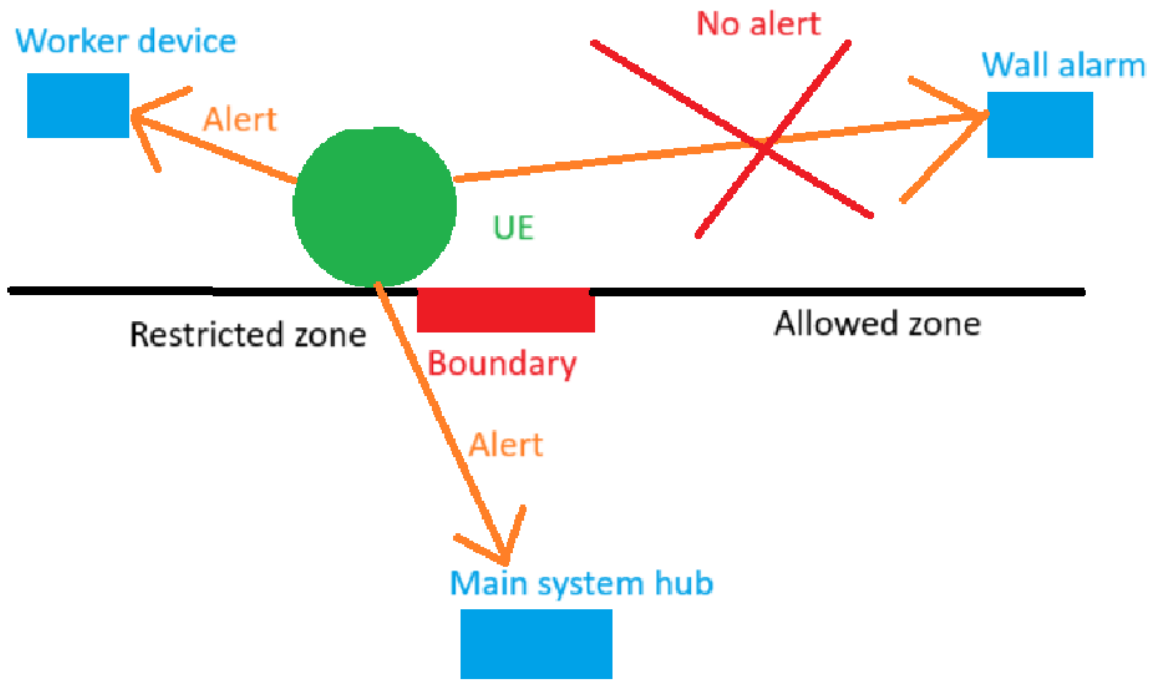
3.3.1. Physical Alerts

Like Jaron and Shailen conceived, there could be tangible, real-world alerts, like audible alarms and flashing lights, which turn on simultaneously on a UE in a restricted zone, in specified places within the restricted zone, and at the location where this product is located. This would be very useful for alerting multiple people at the same time, but might not be very successful in an environment which already has a lot of noise/light pollution. Moreover, it may cost a lot of money to initially set up if there are not existing speakers/lights in the workspace, and would have to be continually updated for each new UE and restricted zone added.



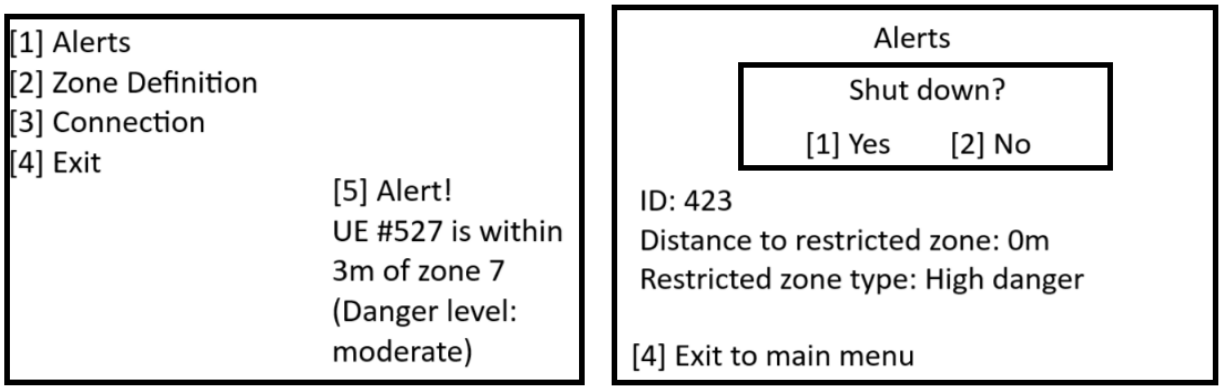
3.3.2. Multi-Party Alerts

Similar to both Brayden and Shailen's ideas, a UE entering a restricted zone could send an alert to multiple people/devices at once, whether they be nearby workers, managers of the workspace, or other UEs in the area. These alerts could be different depending on the devices that they are being sent to, and different UE and boundary combinations could result in different quantities of alerts being sent out. This would help to get everyone affected by an errant UE on the same page, and would resultingly likely improve safety. However, it would require a lot of bandwidth to accomplish efficiently, and may cause undue panic in the workspace over a small event because of the sheer number of alerts being sent simultaneously.



3.3.3. Variable Urgency Alerts

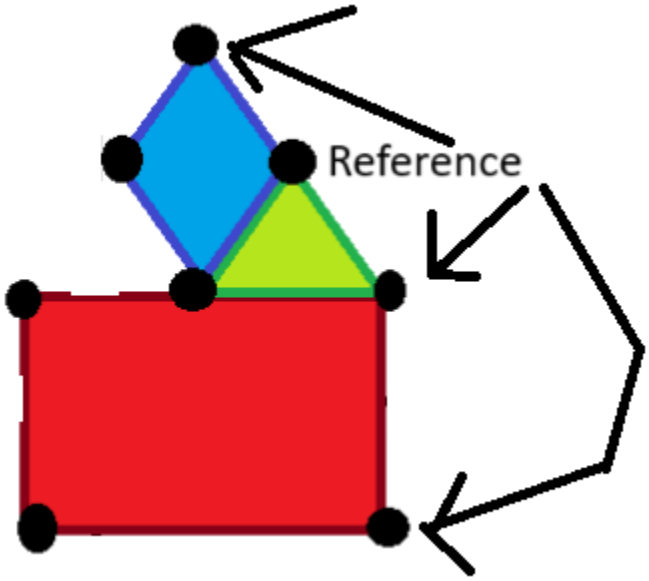
Based primarily on Nick and Brayden's concepts, alerts involving different UEs entering different degrees of restricted zone should logically give different alerts. While some UEs may be very dangerous in some areas, and should instantly prompt a system operator for shut down upon entering a restricted zone, others will not realistically cause much damage, and should not send alerts to disturb the work of everyone around them. This concept also involves different degrees of alert being sent to the same devices; for example, a system operator may receive a small notification for a non-crucial alert, but may also have a full window pop-up requiring necessary action for a crucial alert, as in Nick's original concept. While this modified concept would help streamline workflow, it may prove to be dangerous for not all alerts to be weighted equally, and may also be difficult to adequately implement.



3.4. Zone Definition

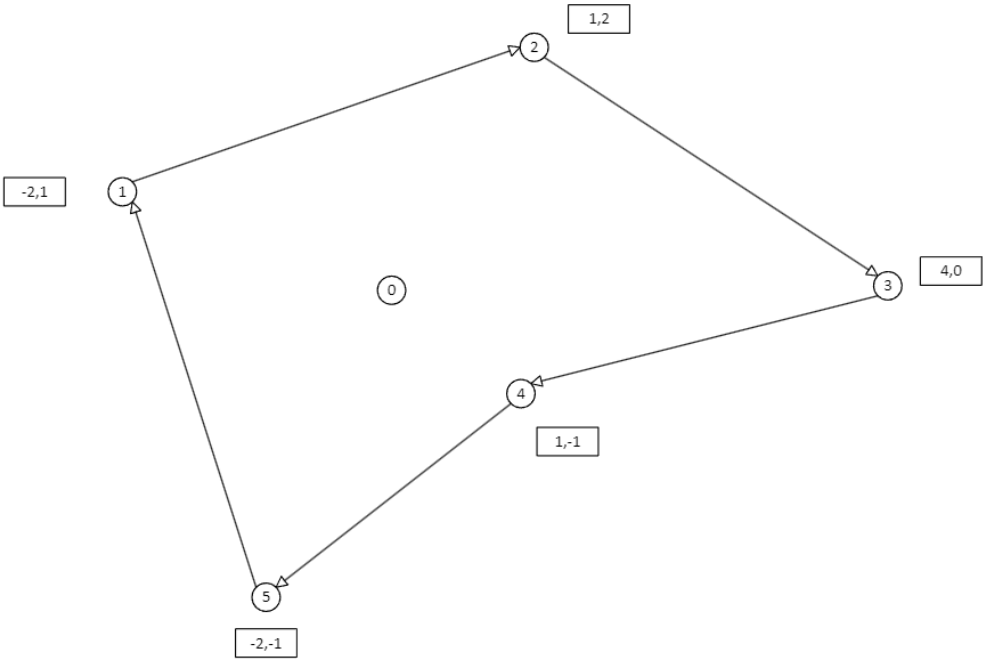
3.4.1. Physical Reference Definition

As mentioned in 3.2.3. Radar Tracking, it can be difficult to precisely measure location without a nearby, real-world reference. As such, like Nick, Shailen, and Jaron all originally conceptualized, there could be a physical marker at each vertex of a restricted zone which reports its position back to the program, and restricted zones can be defined using those points to build lines between. While this would likely be very precise, it would also cost a lot of money to set up, would not work in the case that the connection subsystem ever stopped working for any time period, and might have trouble maintaining the boundaries of a restricted area if the location of a real-world reference changes.



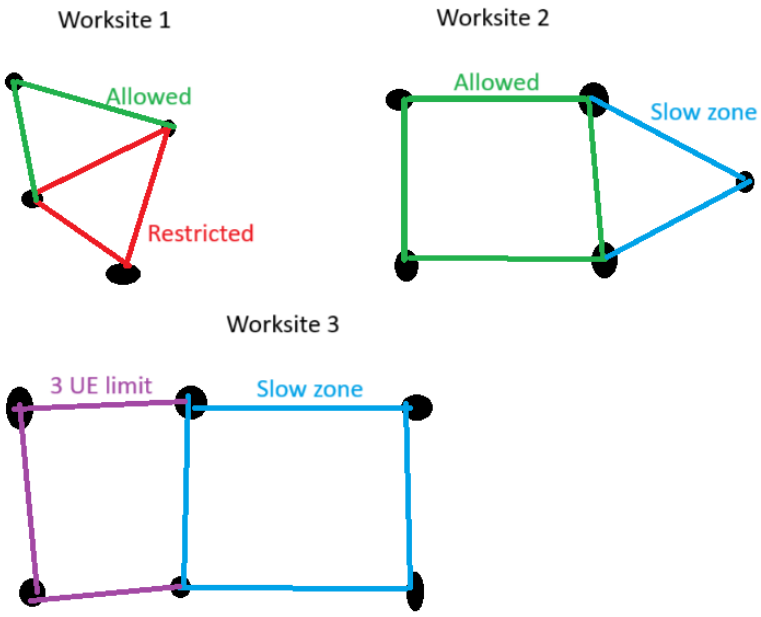
3.4.2. Cartesian Definition

Like Jaron and Nick both mentioned, the Cartesian system of coordinates would be a very useful tool to define the position of restricted zone boundaries, and would allow for simple calculations to discover distances between boundaries and UEs. With a function in the GUI for users to define restricted zones with points they place on a Cartesian plane, it would also be easy to add or modify zones quickly and without any overhead costs. Moreover, it is easy to extend it to a 3rd dimension, which would be useful for any workplace which has a vertical component to its zone restriction requirements. However, this concept is only a model of the real-world, and does not have any intrinsic connections to it, meaning that any changes to the workspace which a user forgets to model, for example, may not offer any alerts even if a UE is in a restricted zone.



3.4.3. Multi-Worksite Definition

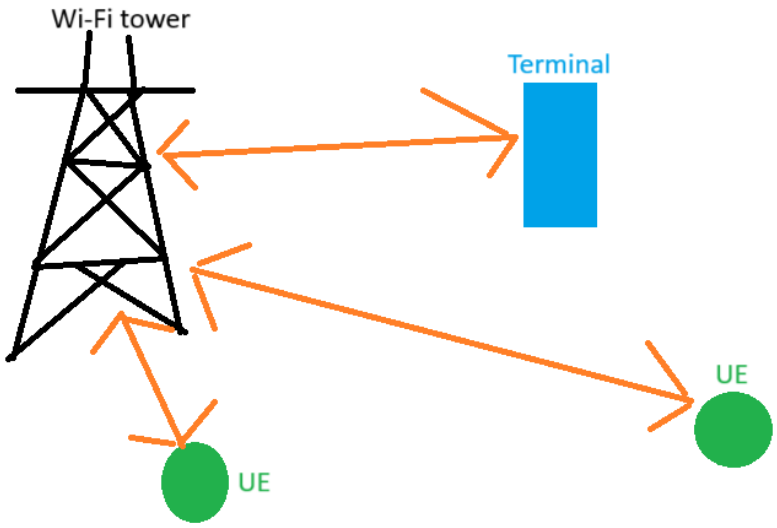
Like Brayden initially considered, a system manager could specify a real-world area for the worksite as a whole, within which each restricted zone could be placed. This concept could be extended to giving that system manager the ability to specify multiple real-world worksites, each of which could have their own restricted zones, and all of which could be modified and monitored from one centralized location. This would be very useful for any enterprise which has multiple campuses, and would reduce the costs for installing terminals to run this product at each worksite. However, keeping track of multiple worksites, especially if each has a lot of vertices, etc. could require a lot of bandwidth, and the GUI for users may become overly complex.



3.5. Connection

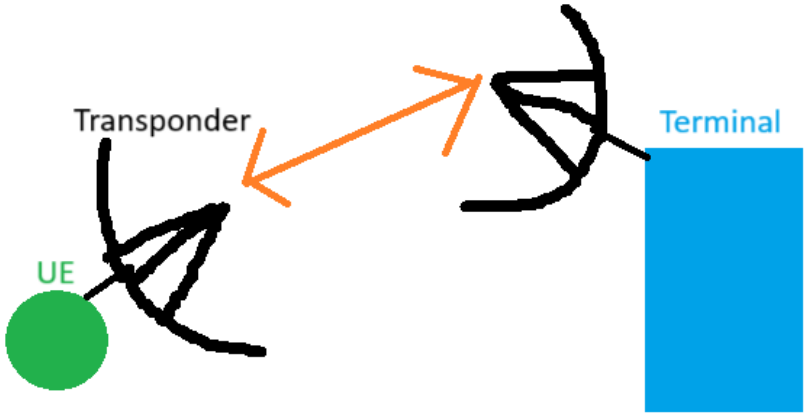
3.5.1. Private Network Connection

As in Shailen, Nick, and Jaron's original concepts, a private network, like an enterprise's Wi-Fi, could be used to connect UEs to the device keeping track of their locations. This is in line with our interpreted needs from the raw data we collected from the first meeting with Shabodi. Assuming the Wi-Fi is consistent, and is able to send and receive a lot of data quickly and accurately, this connection concept could be very secure and efficient with the right network provider. However, it would require that all involved devices have the ability to connect to Wi-Fi, which might incur extra overhead costs, and would also not work if Wi-Fi was not accessible by all devices, for whatever reason.



3.5.2. Radio Connection

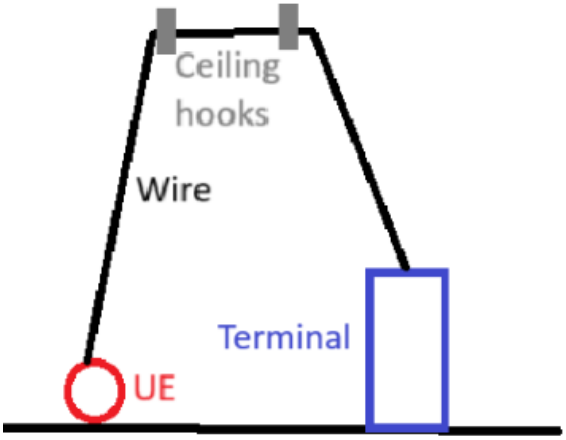
Similar to walkie-talkies, UEs could communicate position information using radio waves. This idea would require that the UE has a localized power source to draw from, but it would almost certainly already have this for its own operation. However, it would also require that the UE has a radio transponder, which would likely have to be set up and which would involve overhead costs for each UE. Moreover, as radio waves are mostly used to carry sound, there is not much technology developed for the transfer of other data using them, and some data might be lost or misinterpreted using this medium. Furthermore, unwanted parties may pick up on these signals, posing a security risk for the user.



3.5.3. Physical Connection

To eliminate most of the uncertainties related to both Wi-Fi and radio connections, a physical wire connection could be used, as Nick originally conceptualized. This would involve running a wire between a terminal and all UEs, and which could run across the ceiling in between these two endpoints in order to stay out of the way. While it would likely be very reliable, it would

also be very prone to having different wires getting tangled together, having UEs run wires over, and having wires get caught on corners. Moreover, there would be a very large initial setup cost, and would also have a large ongoing maintenance cost for all of the mentioned issues.



4. Global Concepts

4.1 Subsystems Ranking Order

4.1.1 GUI System Prioritization

GUI						
Criteria / GUI System	Feasibility (5)	Compatibility (2)	Ease of Use (Implementation) (2)	Adaptibility (1)	Uniqueness (1)	Score
Alert centered GUI	3	2	1	1	1	23
UE Position-Centered GUI	1	3	3	3	3	23
General GUI	2	1	2	2	2	20

4.1.2 Location Tracking Systems Prioritization

	Location Tracking						
Criteria / Location Tracking System	Feasibility (5)	Precision (2)	Compatibility (2)	Ease of Use (Implementation) (2)	Adaptibility (1)	Uniqueness (1)	Score
Variable Time Tracking	2	2	3	2	3	2	29
Boundary Tracking	3	1	2	3	1	1	29
Radar Tracking	1	3	1	1	2	3	20

4.1.3 Alert Systems Prioritization

Alerts							
Criteria / Alert System	Feasibility (5)	Cost (2)	Compatibility (2)	Ease of Use (Implementation) (2)	Adaptibility (1)	Uniqueness (1)	Score
Physical Alerts	1	1	1	1	1	3	13
Multi-Party Alerts	2	2	2	3	2	2	24
Variable Urgency Alerts	3	3	3	2	3	1	29

4.1.4 Zone Definition Systems Prioritization

Criteria / Zone Definition System	Feasibility (5)	Cost (2)	Compatibility (2)	User Friendly Area Selection (2)	Ease of Use (Implementation) (2)	Adaptibility (1)	Uniqueness (1)	Score
Physical Reference Definition	2	1	1	1	1	1	2	21
Cartesian Definition	3	3	3	3	3	3	1	43
Multi-Worksite Definition	1	2	2	2	2	2	3	26

4.1.5 Connection Systems Prioritization

	Connection							
Criteria / Connection System	Feasibility (5)	Maintainability (2)	Compatibility (2)	Ease of Use (Implementation) (2)	Latency (2)	Adaptibility (1)	Uniqueness (1)	Score
Private Network Connection	3	3	3	3	2	3	2	42
Radio Connection	1	2	2	2	1	2	3	24
Physical Connection	2	1	1	1	3	1	1	24

4.2 Three Solutions

4.2.1 First Solution

The first conceptual system uses the alert centered GUI, boundary tracking location subsystem, variable urgency alerts, cartesian definition and a private network connection. These subsystems work well in synergy with each other, for example the alert-based GUI is best used with a more comprehensive alert subsystem like the variable urgency alerts. This system in specific is also somewhat feasible as many of the different ideas for this project would not be easily done in the timeframe given. This system highlights the safety of the user with their limited potential for error occurring.

4.2.2 Second Solution

The second conceptual system uses the UE position centered GUI, variable time tracking, multi-party alerts, physical reference definition and a private network connection. These subsystems together would create a very functional product if executed properly which would be difficult to do in the time frame given. Specifically for the physical reference definition the end user would need to purchase more items for the entire product to be functional, which is undesired.

4.2.3 Third Solution

The third system uses the general GUI, variable time tracking, variable urgency alerts, multi-worksite definition and a private connection. This system is quite comprehensive and would most likely require more time for the group to make a functional application.

5.1 Chosen Global Concept

The best global concept for this project would be the first one stated. This design uses the alert centered GUI, boundary tracking location subsystem, variable urgency alerts, boundary tracking location subsystem, variable urgency alerts, cartesian definition and a private network connection. This design itself was chosen mostly due to its simplistic nature. For most of the design concepts for this project there is not enough time to fully complete. The simplicity of this design makes it much more realistic with the time allotted. The fact that this design is simplistic is also part of the drawback of this design; it is not as comprehensive as the others, but it does cover the requirements necessary. Different concepts like physical reference definition, physical alerts and radar tracking were not selected due to the extra cost and designing necessary to bring these concepts to fruition. While these are all good ideas, it is important to be realistic about the amount of work which the group can get done in the time given. These different systems picked will also work well together as a whole, with different parts of each system complementing each other.

6. Conclusion

Our project's main goal was to create an effective and safe system for monitoring UEs and controlling restricted areas. Real-time alerts are given priority in our alert-centered GUI to stop boundary breaches, even though it may obscure other features like zone management and location tracking. Though it lacks features like UE-UE collision detection, which will be a focus in future iterations, the boundary tracking subsystem simplifies monitoring. We added variable urgency alerts and a private network connection for stability, but scaling and prioritizing alerts are still issues. To increase the efficiency and adaptability of the system going forward, we will concentrate on UE collision detection, mobile integration, automated zone updates, and enhancing connection reliability.