Alex Grommesh
Colin Miller
Othman Al Taie
Preston Beachum
Zach Kaiser

ZOPAC

What don't I know about the system

What data is being received from the endpoints in real time?

How will we monitor endpoint connectivity?

How do we provide reliability if one of the endpoints goes offline?

What config file type will be uploaded to configure each endpoint?

How will we keep Raspberry PI OS fresh?

How will the endpoint notify the server if it goes offline? (Through data on the server)

What data is being received from the wifi signals? Are there associated values, or simply the strength of the signal?

What to do if the device positions get changed without updating the system

How will the heatmap be configured to portray the density of people in an area?

How will we display endpoint health through our UI?

How will we change the shape of the heatmap to correctly fill out a certain floor plan? (If our project is on display at Trask but was tested in a different room, how do we make the program update automatically?) What file type for the floorplan will the user upload?

Uploading the wrong file type may crash the program with improper code.

What information is in “best-known”? Online/offline? Strength of signal? Timestamp?

What qualifies as the best known status? -> The most recent scan with at least 3 end point signals?

Is any encrypted protocol acceptable? Are there any that we are not allowed to use?

How will we verify that the correct endpoints are connected to the server?

How will the application handle static WIFI signals?

How is the WIFI, MAC, and RSSI data going to be presented?

What data is computed upon

Data must be sent to the database and inspected using HTTPS/TLS when transmitted

User can upload a floorplan image/coordinates, and heat map will be applied to that image

When the endpoints are configured, the interface must show the endpoints on the map

Must show the “best-known” status of the endpoints

When the information is available, the location proximity must be shown as a heat map

Instead of best known status, it should be current status used to confirm the system health

What can go wrong with the data or the computation

What to do if there is a data transitioning error
What to do if the time does not match correctly for scan records
What to do if the data cannot be transmitted correctly
What to do if the web application cannot access the server
What to do if the endpoint and/or client cannot connect to server
What to do if the database is down
What to do if the location of an endpoint cannot be accessed

What will happen if some of the data is missing? For instance, the server only receives signals from 2 of the 4 end points?

What if the scanners are offline or have some other issue while the OS is booting and it can’t run on startup?

Domain Model

Model (noun)	Fields
User	email (str), password (str/bytes), salt (int)
EndPoint	x-position (int), y-position (int), z-position (int), id (str or int), status_code (int), status_msg (str)

WifiScan	Timestamp (timestamp), rssi (float), MAC (str/bytes), **EndPoint***		
DevicePosition	Timestamp (timestamp), X-position (int), y-position (int), z-position (int)		
Heatmap	Not sure yet		
Preferences	Language (str), darkmode (boolean), **User***		
API specification			
Client			
Component Name	**Responsibility**	**Call server API**	**Data Needed**
loginStatus	Displays login status with user credentials	FALSE	validateUser == True
validateUser	Validates user credentials before using the system	TRUE	User.credentials
unauthUser	Displays an error message for an unauthorized user	FALSE	validateUser == True
endpointStatus	Displays endpoint status based on data	TRUE	endpoint.status
floorMapChange	Changes floor map based on the file upload	TRUE	client.floormap
accessibilityOptions	Applies alternative text to images and video	FALSE	accessibilityButton == True
Server			
Component Name	**Responsiblity**	**Call server API**	**Data needed**
ping	Creates a query for location and time stamp for coordinates	TRUE	location, timestamp
status_check	Monitor endpoint health	TRUE	endpoint.status
detect_restore	Detects database restarts and restores last valid state of user progress	TRUE	user_progess, db_status
check_time	End session after certain amount of time of inactivity, track last activity timestamp, manage session	TRUE	last_activity, session_length
Server API			
Function	**Signature**		
CREATE	createUser(email, password)		
READ	getEndpointStatus(endpointID)		
READ	getHeatmap(floorplanID)		
UPDATE	updateFloorMap(file)		
DELETE	removeEndpoint(endpointID)		
Endpoint			
Component Name	**Responsiblity**	**Call Server API**	**Data Needed**
config	Load the configuration data of the endpoint from a local file	FALSE	endpoint_ID, server_URL, API_key
scanner	Captures Wi-Fi scan data and packages it	FALSE	MAC, RSSI, timestamp
health_monitor	Detects the devices current health	FALSE	Dongle_presence, disk_usage, CPU_stats, memory_stats, online_status
local_log	Contains operational events	FALSE	Auth_events, errors
auth_system	Verifies endpoint successfully connects to the server	TRUE	API_key