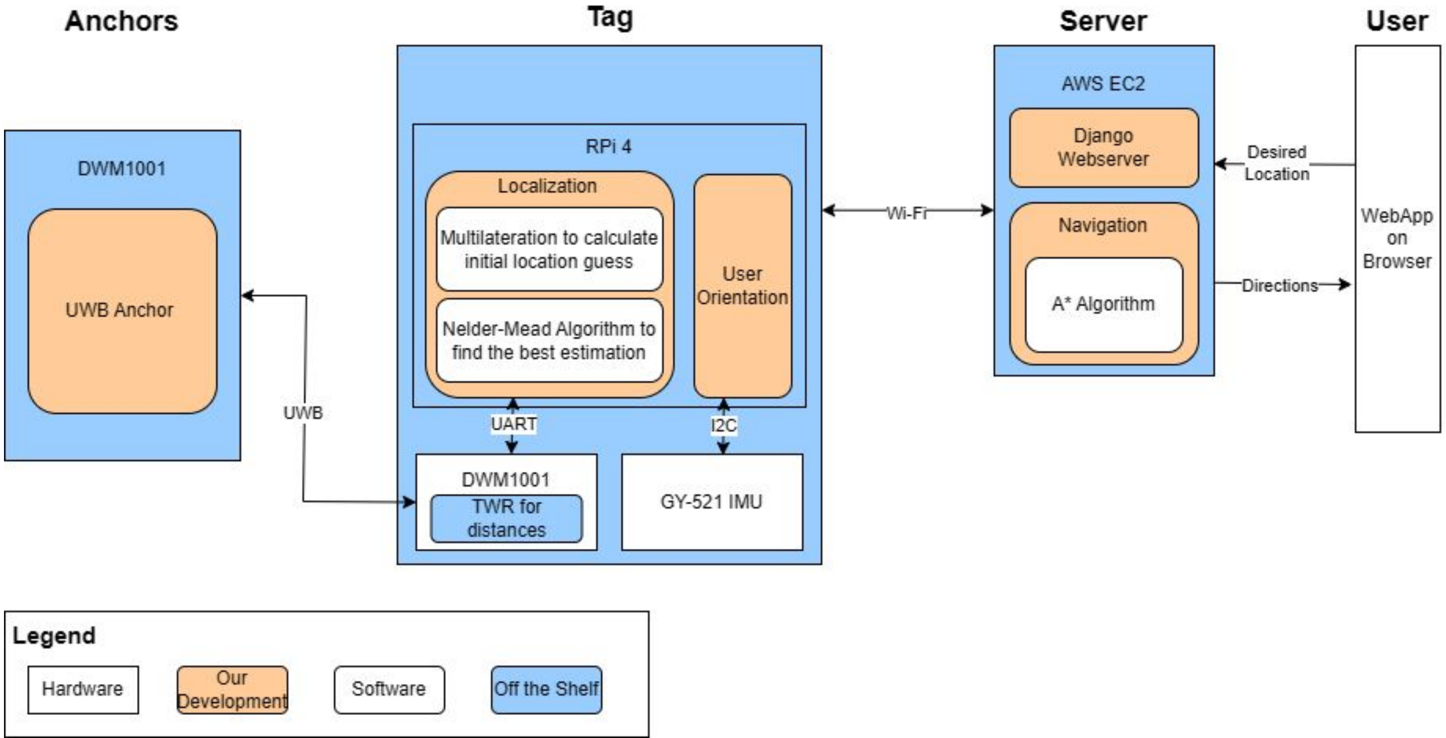
Use Case Requirements

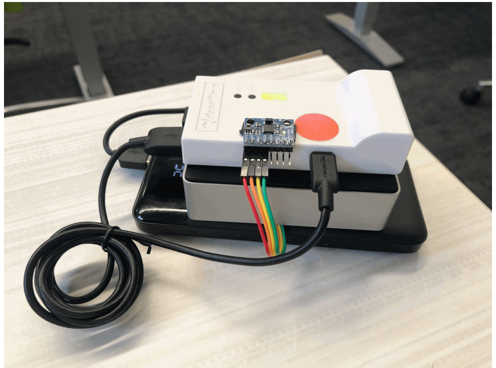
Requirement	**Use Case Requirements**	**Design Requirements**
Accurate localization	**< 1 meter**	Individual distance measurements **< 0.23 meter accuracy**
Battery life of device	**> 4 hours**	Capacity **>10 mAh , 5 V** battery Power consumption **< 12.5 watts**
Responsive tracking	**> 2 Hz** update frequency	Distance acquisition and localization algorithms **< 500 ms**
Device Price	**< $75**	**< $75**
Infrastructure Price	**< $100 per hallway**	UWB transceivers **> 25 m** range

Solution Approach



Complete Solution

The tag shows the estimated position on a map to the user via a companion web app



Testing Procedures

Accuracy Tests

- Measure actual positions and distances: A laser measuring tool with millimeter precision
- Find estimated position using pixel offsets on the map
- Calculate the distances between actual positions and estimated positions

Testing, Verification, Validation

Test	**Inputs**	**Passing**	**Result**
Range of Anchors and Tags	Maximum communication range within a closed space	**> 25 m**	**34 m**
Distance Measurement Accuracy	Average distance accuracy between two DWM1001's	**< 0.23 m**	**0.15 m**
Localization Accuracy	Compare predicted location with the actual location	**< 1 m**	**0.2 m**

Latency Test Procedure

Software Latency

- Use Python Time library to measure the time for algorithms and web server updates

Total Latency

- Record a video of the user moving
- Measure time difference between movement and display on browser

Testing, Verification, Validation

Test	Inputs	Passing	Result
Localization Precision	Maximum fluctuations in predicted location	**< 0.5 m**	**2.1 m**
Heading Accuracy	Average difference of the angle of the user's estimated orientation compared to reality	**< 20°**	**To Be Determined**
Battery life of tag	Measure battery life of device	**> 4 hours**	**10 hours**

Testing, Verification, Validation

Test	**Inputs**	**Passing**	**Result**
Position Update Latency	Measure latency of distance calculating algorithm	**< 500 ms**	**20 ms**
Distance Update Frequency	Measure frequency to get new distance value	**> 2 Hz**	**10.1 Hz**
Tag to WebApp Latency	Latency from sending information to webapp	**< 250 ms**	**67 ms**
Total Latency	Total latency from changing position in real world to getting reflected on the map	**< 2 sec**	**0.84 sec**

Testing, Verification, Validation

Test	**Inputs**	**Passing**	**Result**
Navigation algorithm	Varying starting and ending locations	Shortest paths found 100% of time	Length of path always <= BFS
Navigation Algorithm Speed	Furthest start/end positions on graph	< 500 ms	125 ms
User Experience	Qualitative feedback from clients for quality of directions	Clients think directions are helpful	4 users provided a rating of 4.5/5.0

Design Tradeoffs

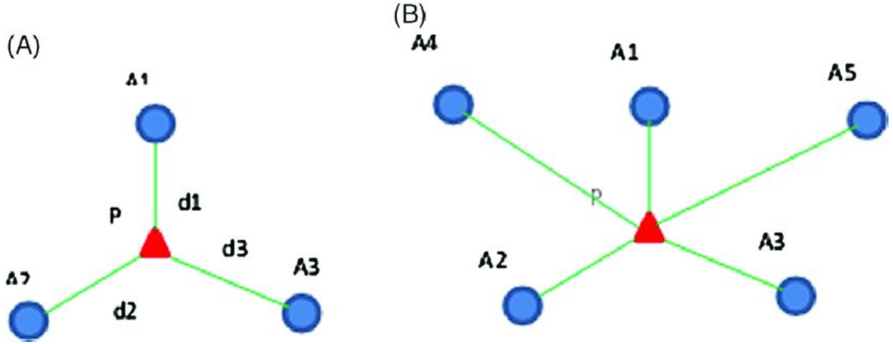
Ultrawideband Device

ESP32 UWB	Raspberry Pi 4
• Lower cost ($39.50) • Lower Operating frequency (160MHz) • No in-built USB-UART	• **Higher cost ($54.99)** • **Higher Operating frequency (1.5GHz)** • **Has in-built USB-UART**

Server Communication Protocol

HTML Post/Requests	Websockets
• 250 ms update time • More intensive	• **67 ms update time** • **Offload work to Redis server**

Design Tradeoffs



Position Estimators

Trilateration	Multilateration
• Uses only 3 anchor readings • Only pick closest 3 anchors	• **Uses >=3 anchor readings (With 4 anchors, accuracy improves ~10cm)**

Position Solvers

Gradient Descent	Nelder-Mead algorithm
• Higher Precision • Minimizes until it converges (stuck at local minimum)	• **Lower Precision** • **Don't have convergence issue (~500 ms faster on Raspberry Pi)**

Project Management

TASKS	PROGRESS	START	END
Jeff			
Django app Setup and models	100%	2/18/24	2/21/24
Frontend display of map + building	100%	2/22/24	2/24/24
Frontend: JS controlled building img in container	100%	2/25/24	3/3/24
A* algorithm, display path on floor	100%	3/11/24	3/17/24
Communication from tag to webapp	100%	3/18/24	3/24/24
Test Websocket and test A*	100%	3/25/24	3/31/24
Navigation Paths	100%	4/1/24	4/7/24
Navigation Directions	100%	4/8/24	4/14/24
Refine UI, Resolve bugs	100%	4/15/24	4/21/24
Weelie			
Test DWM1001-Dev boards	100%	2/18/24	2/24/24
Testing network of Anchors/Tag	100%	2/25/24	3/3/24
Localization Algorithms	100%	3/11/24	3/17/24
Communication from tag to webapp	100%	3/18/24	3/24/24
Improve localization accuracy	100%	3/25/24	3/31/24
Improve anchor network and localization latency	100%	4/1/24	4/7/24
Localization with larger network	100%	4/8/24	4/14/24
Ifeanyi			
Test DWM1001-Dev boards	100%	2/18/24	2/24/24
DWM1001 Gateway w/Rpi	100%	2/25/24	3/3/24
Program IMU for the Tag	100%	3/11/24	4/7/24
Gradient Descent algorithm	100%	3/25/24	3/31/24
Everyone			
Testing inside a Hallway/Room	100%	3/25/24	3/31/24
Testing Navigation System	100%	4/8/24	4/14/24
Slack	100%	4/15/24	4/21/24
Final Testing and Troubleshooting	0%	4/22/24	5/1/24
Final Deliverables	0%	4/22/24	5/4/24