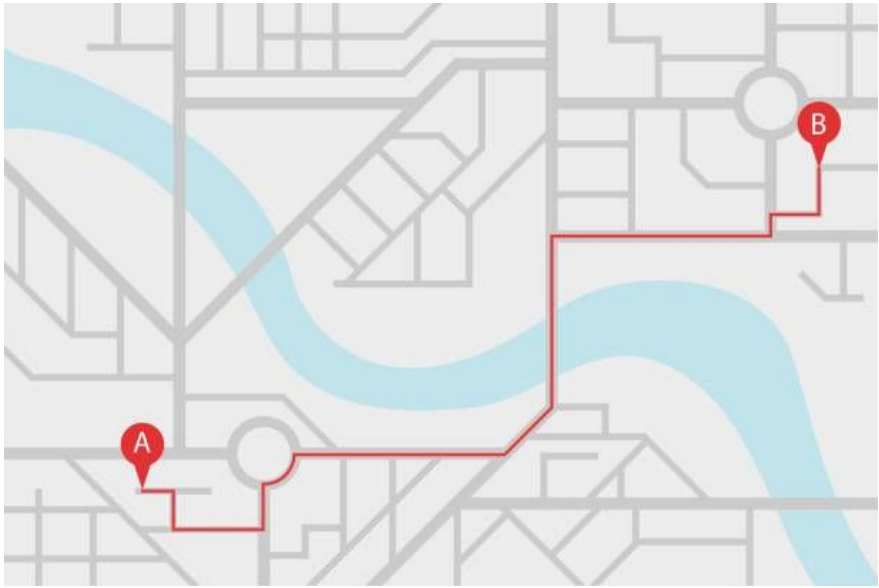
Use Case

- User is a visually impaired person
- Wants to go from point A to point B
- Ideally, only wants routes with “blind-friendly” crosswalks
  - Crosswalks with audio cues which signal when it is safe to cross
  - Text-to-speech, beeping, etc.
- Aid the visually impaired person in learning a new area
- Rescope: Only around area immediately surrounding CMU



Use-case Requirements

- Update user frequently so they are informed about where they are going
- Identification of user location within **1 meter** of actual location
- Battery life of **16 hrs**
- Reliability: user should be confident in directions/information given
- Latency: should return response (ie. directions) **<1s** after coordinates of user is given
- Weight: **<1kg**

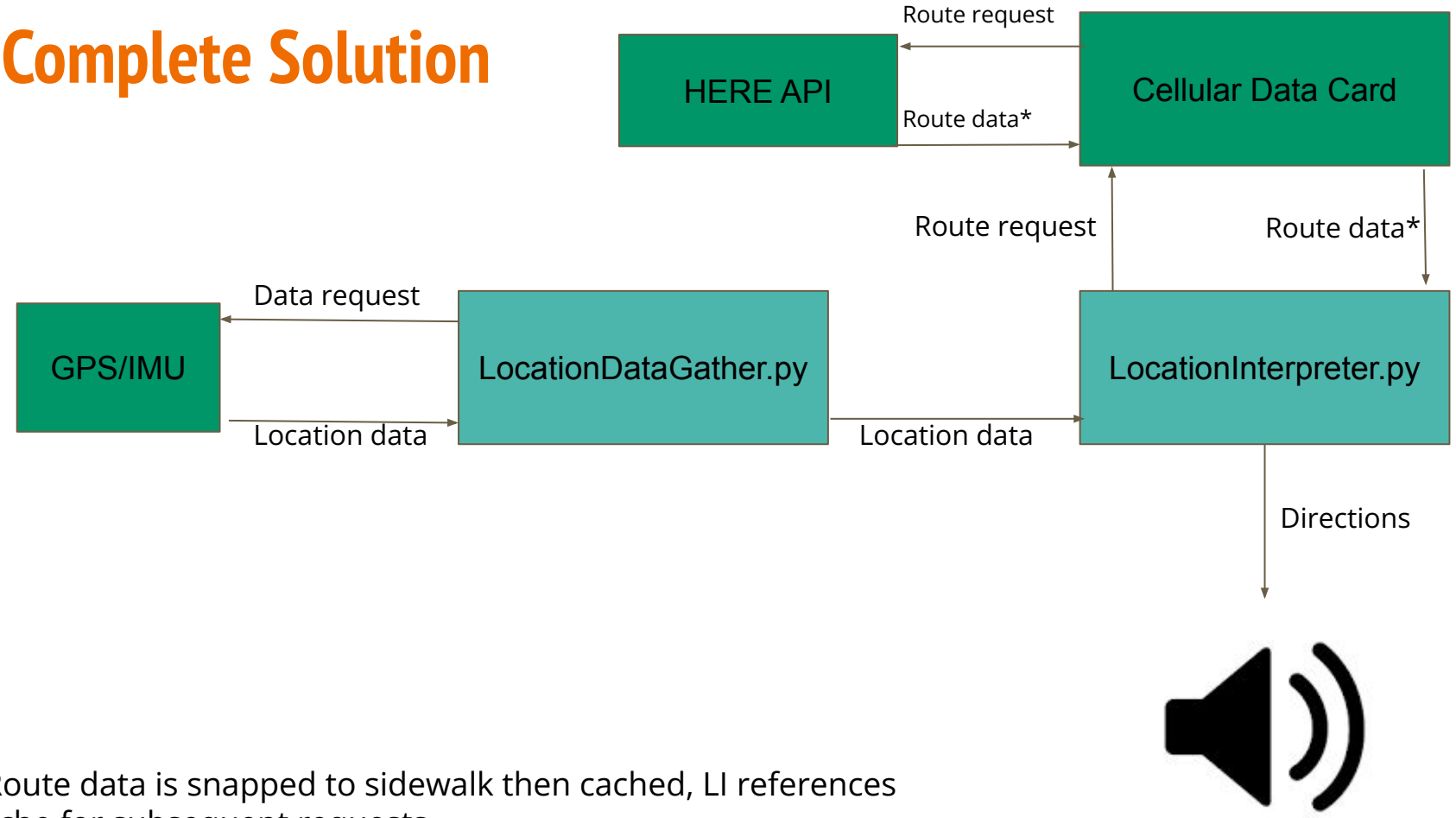
Solutions Approach

- Product: wearable device which gives user route from point A to point B
  - Route will only contain “blind-friendly” crosswalks
  - If user deviates from route, notify user if they attempt to cross “blind-unfriendly” crosswalk
  - If user deviates from route, reroute them based on direction they are currently walking in
- Front-end:
  - Combination of inertial measurement unit and GPS locator
    - ~~I2C~~ UART Serial Communication
  - Audio feedback via wired earbuds
    - 3.5mm audio jack
  - Wireless communication to ~~Maps~~ HERE API
    - Via cellular data network card
    - I2C Serial Communication

Solutions Approach

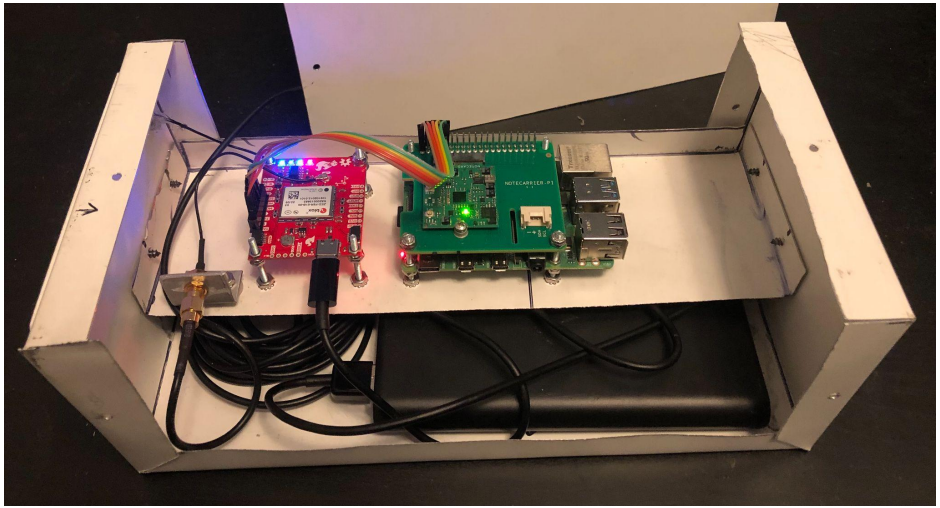
- Backend:
  - Will keep a database of “blind-unfriendly” intersections
  - ~~Find possible routes from A to B, and filter after for blind-friendly routes~~ (Taken care of by HERE API)
  - Find coordinates of next intersection/crosswalk to turn at, calculate distance between user current location
  - Give feedback to frontend to direct to user
  - Use ~~Google Maps and Overpass API~~ HERE API to assist with route planning and intersection mapping

Complete Solution



*Route data is snapped to sidewalk then cached, LI references cache for subsequent requests

Complete Solution (Hardware)



TODO: add on/off switch, charging port, aux port

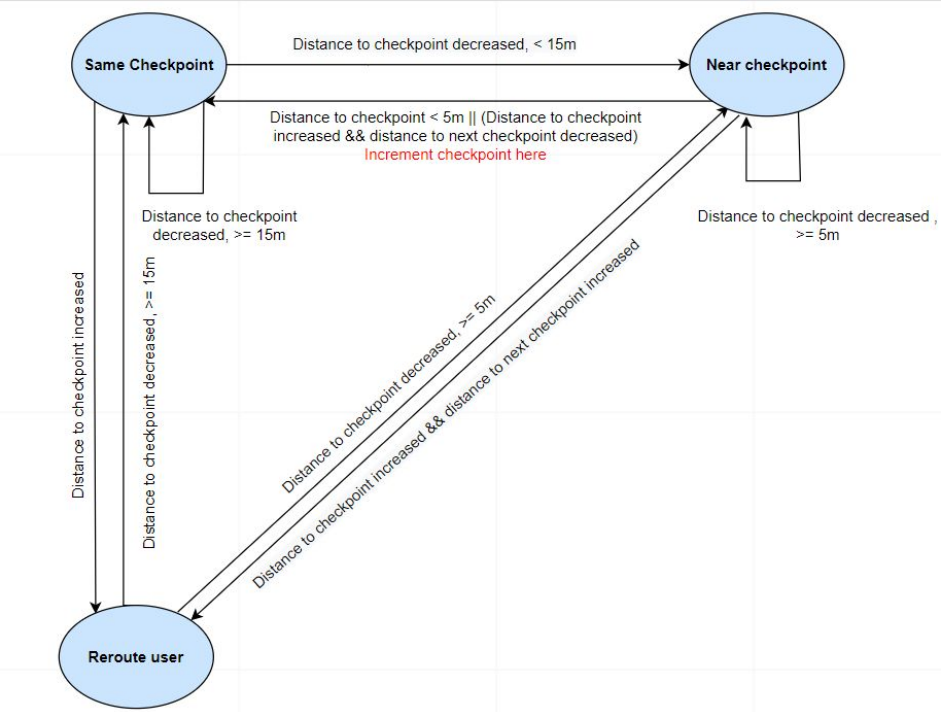
TODO: Incorporate into full wearable device (currently using a bag now)

Complete Solution (Software)

- Translation of coordinates (Lat, Lng) between two points to distance: Haversine Function
  - $$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$
- Call to HERE API to get direction from user location to intended destination
  - Returned data is cached, cache is accessed for subsequent requests, reducing monetary cost and latency
- TODO: “Snap”/translate road coordinates to account for which side of the street user is on

Complete Solution (State Transitions)

- API data consists of checkpoints
- State transitions required to know:
  - when to proceed to next checkpoint, or
  - when to reroute the user because they are deviating from original path



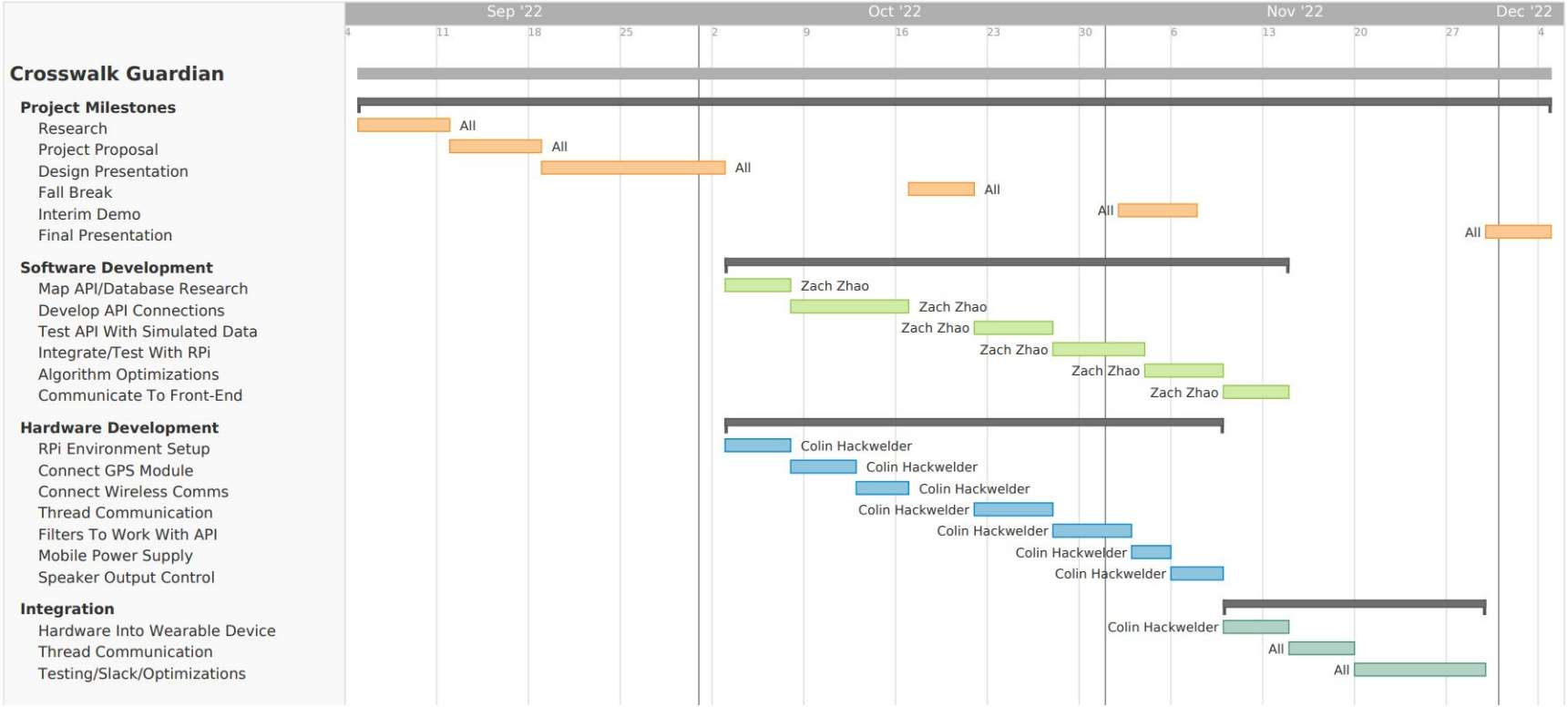
Testing, Verification, Metrics

Requirement	Test	Goal	Our Measurement
Battery Life	Monitor power consumed during normal operation	16 hours	Peak: 1.1 A @ 4.93 V 24 hours
Unfriendly crosswalk detection	Stand at various distances <10m away from unfriendly crosswalk	What percentage of the time does system detect unfriendly crosswalk	TBD
Latency	Measure time that it takes for system to provide feedback	< 1 second	Without re-route: 0.0046 s During re-route: 1.72 s

Testing, Verification, Metrics

Requirement	Test	Goal	Our Measurement
Location Accuracy	Walk at known coordinates and measure accuracy	< 1 meter deviation from actual location	Ideal conditions (out in the open): 0.9m - 1m Urban conditions (against large buildings): 5m
Weight	Scale	< 1 kg	0.85 kg
User experience	Walk a full route planned by system	Determine if system can provide accurate and safe route/directions	TBD

Schedule


Sep '22
Oct '22
Nov '22
Dec '22
4
11
18
25
2
9
16
23
30
6
13
20
27
4
Crosswalk Guardian
Project Milestones
Research
Project Proposal
Design Presentation
Fall Break
Interim Demo
Final Presentation
Software Development
Map API/Database Research
Develop API Connections
Test API With Simulated Data
Integrate/Test With RPi
Algorithm Optimizations
Communicate To Front-End
Hardware Development
RPi Environment Setup
Connect GPS Module
Connect Wireless Comms
Thread Communication
Filters To Work With API
Mobile Power Supply
Speaker Output Control
Integration
Hardware Into Wearable Device
Thread Communication
Testing/Slack/Optimizations
All
All
All
All
All
All
Zach Zhao
Zach Zhao
Zach Zhao
Zach Zhao
Zach Zhao
Zach Zhao
Colin Hackwelder
Colin Hackwelder
Colin Hackwelder
Colin Hackwelder
Colin Hackwelder
Colin Hackwelder
Colin Hackwelder
Colin Hackwelder
All
All

Challenges/Trade-offs

- Location accuracy falls off in an urban environment
  - IMU sensor fusion was supposed to be the solution
  - IMU/GPS unit is optimized for faster moving applications (cars)
  - In reality IMU does not help our situation
  - Trade-off: To gain IMU accuracy, must increase device rigidity (increase weight)
- No APIs support sidewalk-specific routes
  - Not all blind-unfriendly intersections need to be avoided
  - Need to gather coordinates of all sidewalk corners
  - Translate route: snap API route to nearest sidewalk
- Smallest street+sidewalk width (one side): ~4.5 meters < 5 meters (our urban accuracy)
  - Cannot reliably determine which side of the street a user is on