EleNA

Elevation-Based Navigation App

Team: The Software Engineers

Team Members:
Casey Ryan
Cody Richter
Sohan Show

Requirements Engineering

Guiding Principles

Functional Requirements

- Authentication
- Map API Connection
- Efficient Pathfinding Algorithm

Non-Functional Requirements

- Usability
- Testability
- Privacy

Full requirements document available here: [link]
(must use UMass account to view)

Why Choose Us?

The User Comes First:

- All data is stored locally and never shared with external services

Lightweight:

- Minimal data is transmitted between server and client, allowing usage in low-connectivity areas

Extensible Across the World

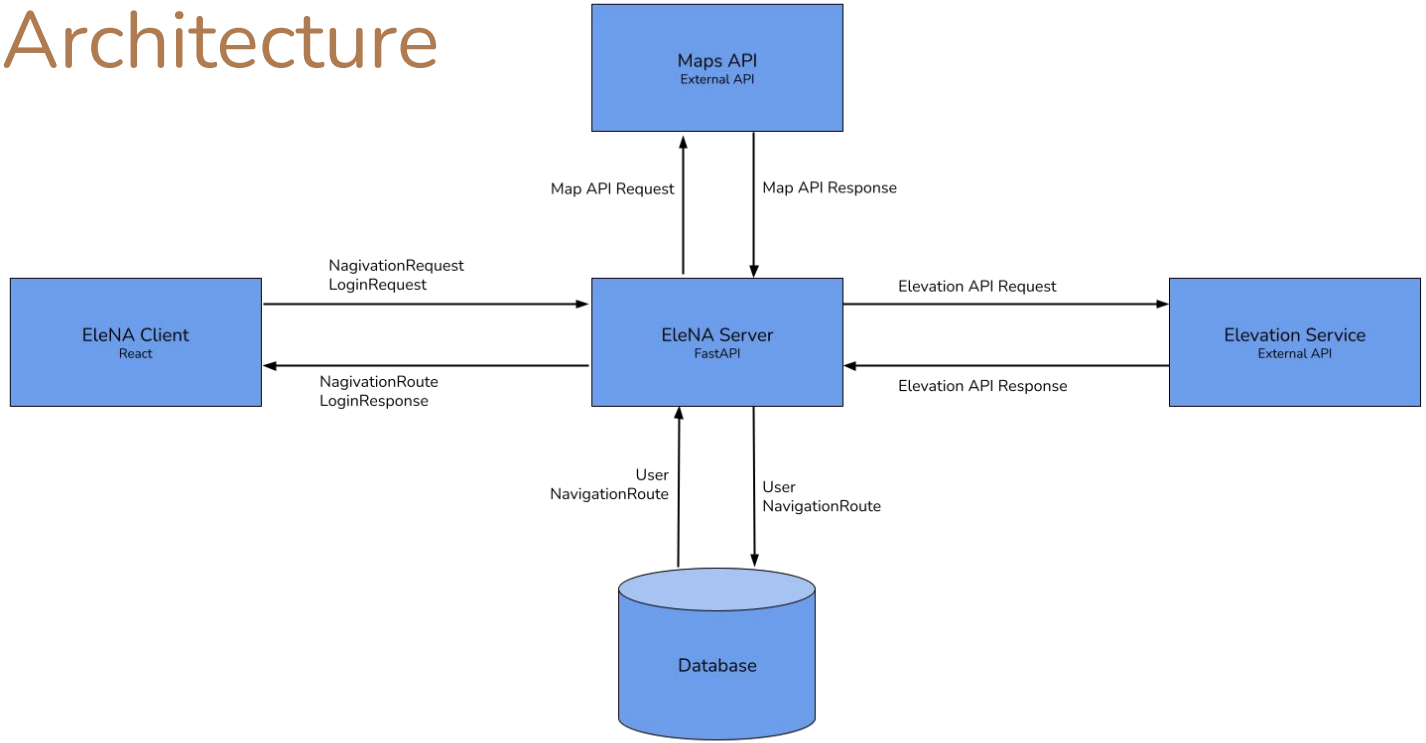
- From mountains to cities, we can create routes
- Available in any country which supports OpenStreetMaps

Project Design/Architecture

Scalability

- Containerization
  - All components of project self-contained and buildable from Dockerfiles.
  - Containerization of all services enables scalable cloud hosting
- Orchestration
  - Docker-compose used to manage multiple containers simultaneously, scalable in the future to large scale orchestration tools like Kubernetes
  - If large load on the system, can spin up additional containers to meet demand.
- Location-Independent Routing
  - Scalability of our system allows for routing in any geographic area, regardless of demand.

Architecture



Data model details available here: [link]
(must use UMass account to view)

Implementation

Backend Implementation

- FastAPI routers used to modularize components
  - Authentication, navigation, and user accounts isolated from each other
- Python docstrings added to all methods and generated in HTML website to maximize understandability
- Pathfinding algorithm does not rely on external API to find create routes between locations

Frontend Implementation

- React components contain minimal code, and folder structure maximizes understandability
- All network calls and dependencies in separate module
- Asynchronous state updates maximize user engagement

Elevation Service

- Created with docker container forked (and modified) from existing Open-Elevation API
- Uses NASA SRTM (topography) data to map each lat-lon pair to an elevation
- Functions as API on localhost to minimize latency from main application

Pathfinding Algorithm/s

- A*
  - Easy to implement
  - Reasonably performant
  - Cost function allows us to handle the elevation preference directly in the algorithm
    - Edge weights must be positive
  - O(|E|)
- Custom Maximize-Elevation Algorithm
  - Longest path problem is NP-Complete, so we need to compromise somewhere
  - Gets true shortest path then does linear scan to find max elevation path in between each shortest path node
    - Max elevation path between nodes limited by depth

Verification and Validation

Testing and Project Testability

- **Backend**: pytest Python testing framework
  - Integration with FastAPI, our backend framework
  - Tests are run in parallel
  - Dependency injection used for routes, so external dependencies and calls can be mocked
  - Pathfinding, authentication, and routers have comprehensive tests

- **Frontend**: React component jest testing
  - Will compare to hardcoded ground truths
  - Code coverage above 90% for key components

Contribution and Development Process

- Develop in separate branches and merge into main only after comprehensive review
  - Another person must review prior to merge, and they must run the changes locally first
  - If a team member contributed to the merge request, they are not allowed to approve it
- Github Issue and Pull Request Templates
  - Ensure all issues and PRs submitted adhere to our standards
- Development Timeline Followed

User Study

After having gone through verification and testing, we gave our application to 5 different people across different majors at Umass Amherst and asked them for their feedback. Here's what they had to say:

- **Junior, Communications Major:** The login page and UI looks very clean. I really like the idea of getting path based on elevation. It works really cool. Maybe if you guys could make an app out of it.
- **Sophomore, Engineering Major:** This is really cool. I liked the part where I could maximize and minimize elevation. Super helpful for my hiking trips. Maybe change the application logo?
- **Senior, CS major:** You guys did a great job here. The UI looks clean and minimalistic, the endpoints seem to be working flawlessly. It would be great if you could add an autocomplete when user types the source and destination.
- **Junior, Biomed major:** This looks so much like google maps. Did you guys use something similar? This is really amazing. The max and min elevation is so cool. Really love it. Maybe make this into an app.
- **Senior Fine Arts major:** I really like the part where it stores my previous routes and gives me max and min elevation. This looks like something that would be super useful during hiking. I really like the UI, it's super clean and easy to use. Can I use it on my phone?

Live Demonstration
(External)

Links and References

- **Project Github Page:** [Link]
- **Project Documents and Diagrams:** [Link]
- **Open-Elevation:** [Link]
- **OpenStreetMap:** [Link]
- **Leaflet API:** [Link]

Thank You For a Great Semester!