MICHIGAN STATE UNIVERSITY

Project Plan Presentation

AWSome Availability Zones

The Capstone Experience

Team Amazon

Huang, Wynton
Chak, Jung
Kim, Iris
Heiner, Jamison
Hood, Jake

Department of Computer Science and Engineering
Michigan State University

Fall 2021

From Students…
…to Professionals

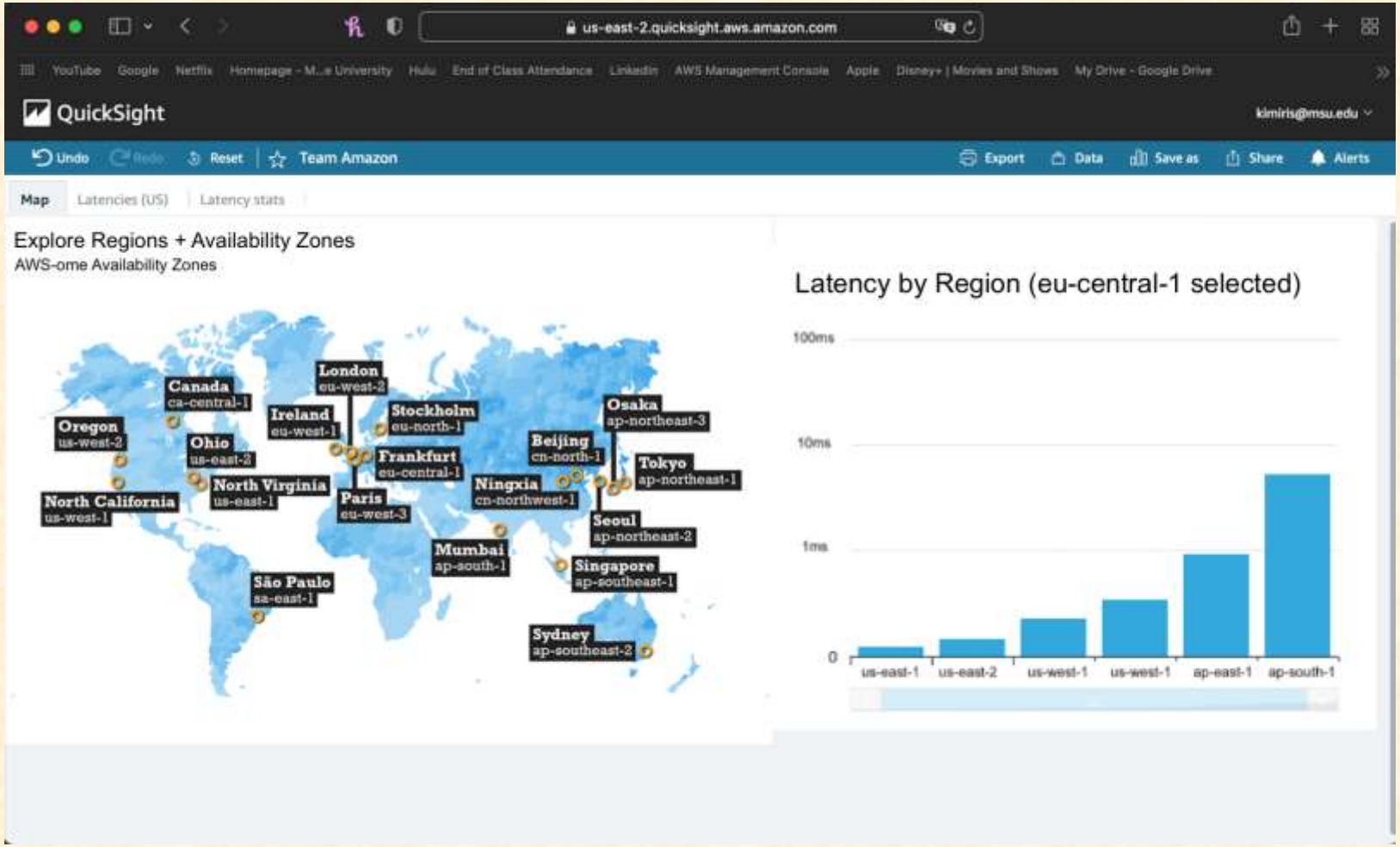
Functional Specifications

- Gather information on network latency between Availability Zones (i.e., data centers)
- Assist users with optimizing network latency in their applications by suggesting AZ pairs
- Interact with data through easy-to-use UI
- Filter through results based on region-based requirements (ex. no AZs in Europe, one must be in Western US, etc.)

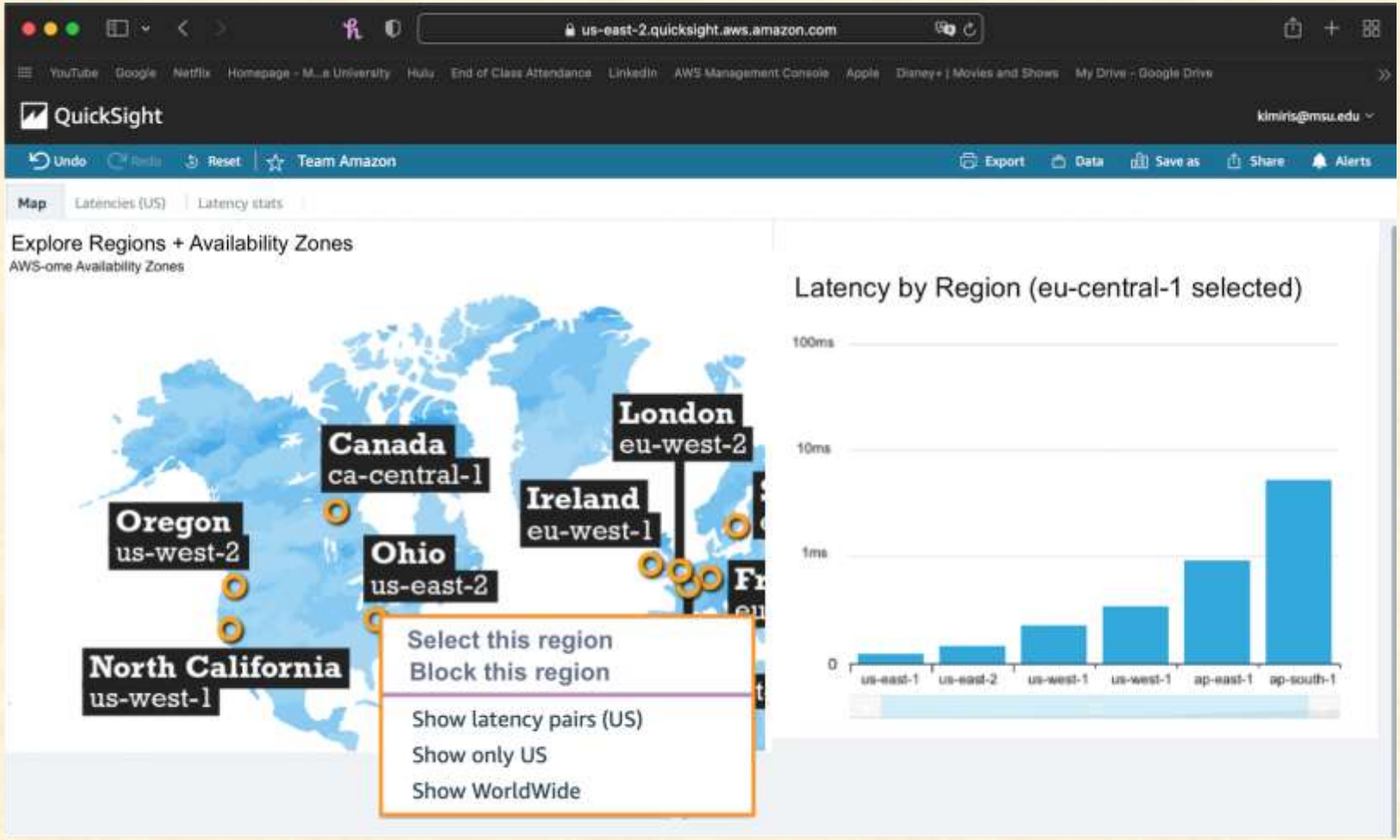
Design Specifications

- User friendly + interactive map showing all Regions in the world. Each Region is clickable and will show more details about each Availability Zone it contains
- Filtering system where users can add, edit, and remove Region-based requirements for AZ placement
- Once a Region is selected, all other Regions are colored on main map webpage based on average network latency
- Statistics webpage for any pair of Availability Zones showing a graph of network latency tests and highlighting important features like average latency, 95^{th}, 99^{th}, and 99.9^{th} percentiles

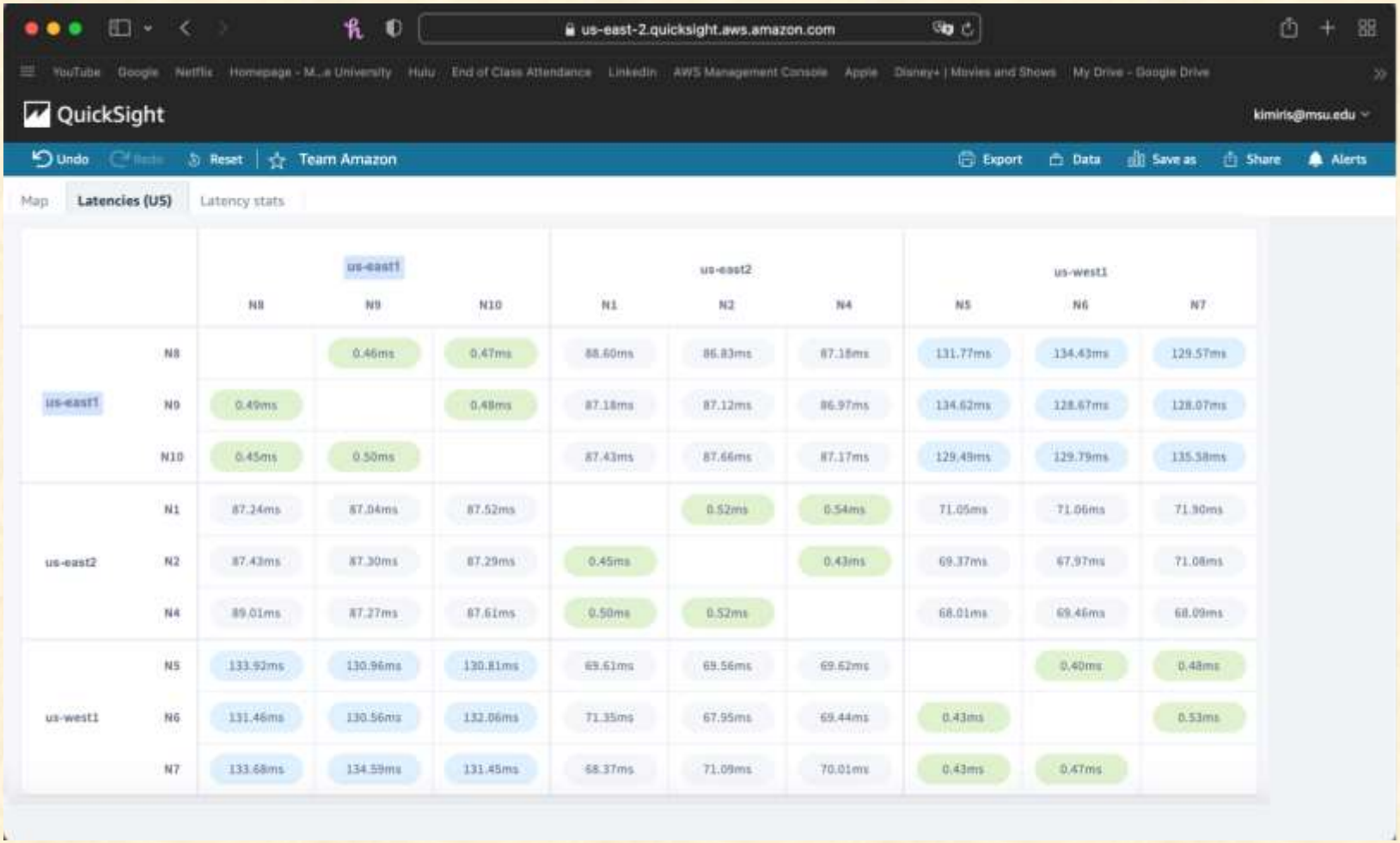
Screen Mockup: Interactive Map



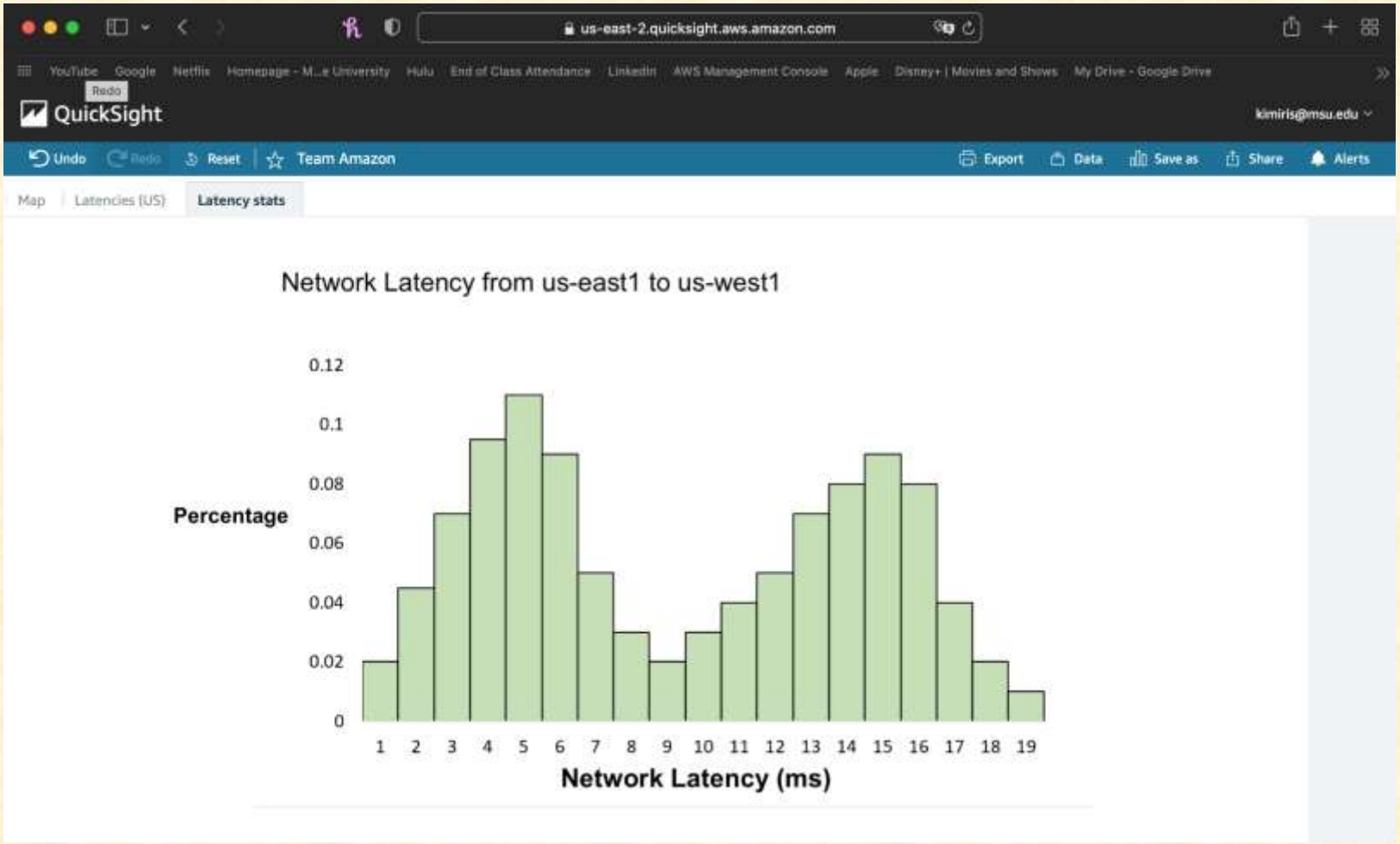
Screen Mockup: Select Region



Screen Mockup: Regional AZ Matchups Page



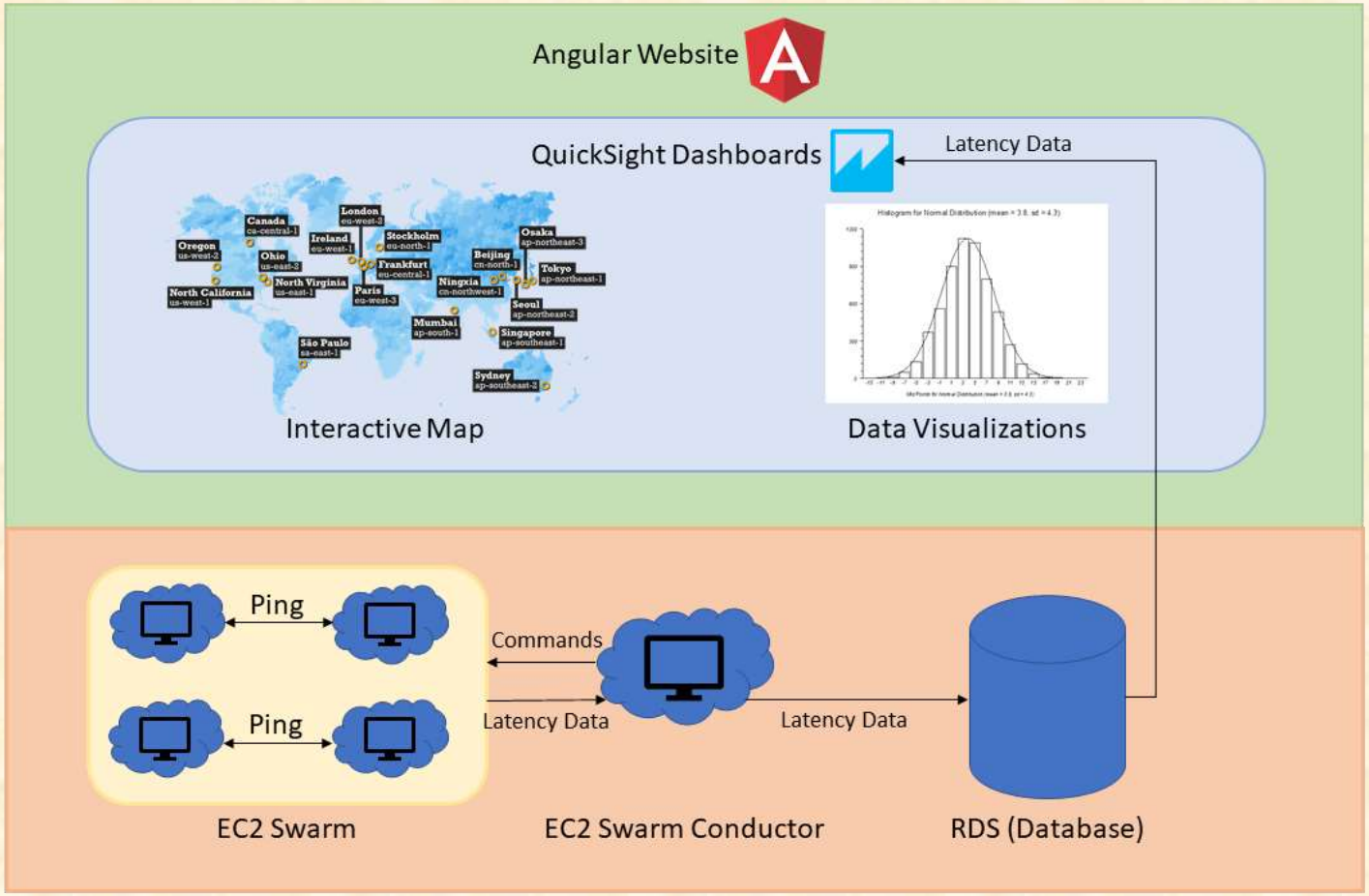
Screen Mockup: Specific AZ Pair Statistics Page



Technical Specifications

- Front End
  - QuickSight Dashboard
  - GitHub Pages
  - Angular web application
- Back End
  - AZ Matching Algorithm
  - ICMP vs. TCP
  - High Precision Timer
  - EC2 automatic deploy

System Architecture



System Components

- Hardware Platforms
  - N/A
- Software Platforms / Technologies
  - Front End
    - Angular, Angular Materials, GitHub Pages
    - AWS Services (IAM, Lambda, API Gateway, QuickSight)
  - Back End
    - AWS CDK, CLI, SDK
    - Amazon TimeSync
    - Chrony, C, TCP, DNS, HR Timer, NTP

Risks

- Back end/Front end Communication
  - The back end database is likely to change over time, and these changes will affect the front end
  - API interface between front and back to control when communication happens
- Timing methods and Precision
  - Latency between AZs is naturally low, so timing latency needs to be precise and consistent
  - Timing using clock ticks for precision, and consistency achieved with time
- IP Address Security
  - Deployment of our detection method requires knowledge of IP addresses, which may be inconsistent and a security risk
  - Deploy Swarm Conductor first
- Unknown QuickSight Functionalities
  - QuickSight might have/lack functionalities that will interfere with our planned display
  - Use and experiment with embedded versions on Angular to create workarounds

Questions?