MEMORIAL UNIVERSITY

Faculty of Engineering and Applied Science
Fall 2024

ENGI 9837 - Software Engineering Capstone

Radar Data Viewer

Project Supervisor: Prof. Reza Shahidi

Group Members:

1. Eialid Ahmed Joy - 202384815
2. Monirul Islam - 202381823
3. Jakaria Hossain - 202293102
4. B M Al-amin Badhon - 202384559
5. Sadia Noshin Shuchi - 202380413

Date: December 13th, 2024

Introduction

Project Overview

- A web-based platform for visualizing HF and X-Band radar data.
- Initial focus: Display radar data files in **B-Scan format**.
- Displays metadata such as transmission frequency, bandwidth, range, and location.
- Future plans: Add support for animations, advanced data analysis (e.g., spectrum and parameter extraction).

Introduction (Cont.)

Project Objective

- Create an intuitive, user-friendly application for radar data visualization.
- Simplify uploading, browsing, and analyzing radar data files.

Project Methodology

Requirements Gathering

- Identify user needs and project objectives.
- Document functional and non-functional requirements.

Planning and Design

- Develop a project plan with timelines and milestones.
- Create system architecture and user interface designs.

Development

- Implement core functionalities in iterative cycles.
- Use modular development to enable scalability and flexibility.

Testing

- Perform unit, integration, and system testing.
- Address bugs and refine features.

Deployment

- Deploy the application on chosen platforms.

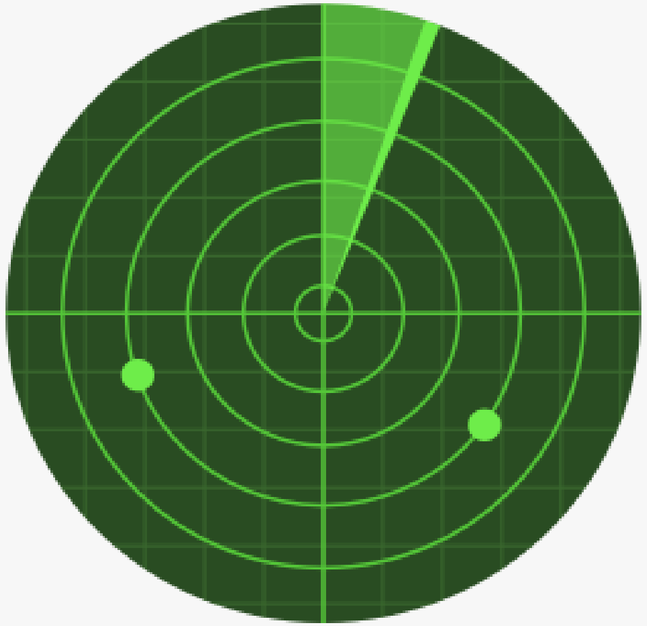
Maintenance

- Monitor system performance and resolve issues.
- Plan and implement advanced features like animations and data analysis in future versions.

MVP

Included Features

- Visualize a single HF radar file in B-Scan format.
- Display key metadata like transmission frequency and acquisition info.

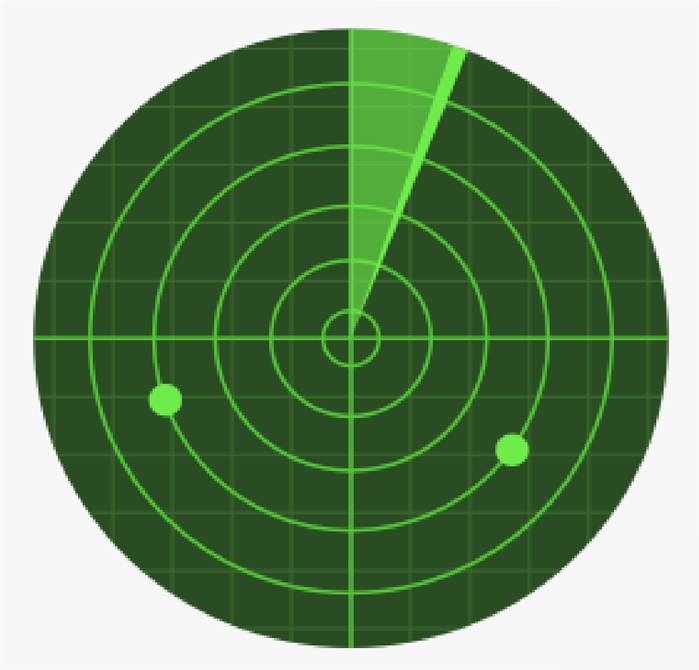


- View multiple files as animations.
- Web-based system accessible on major OS platforms.

Final Product

Additional Features

- User login and registration
- Token Validation



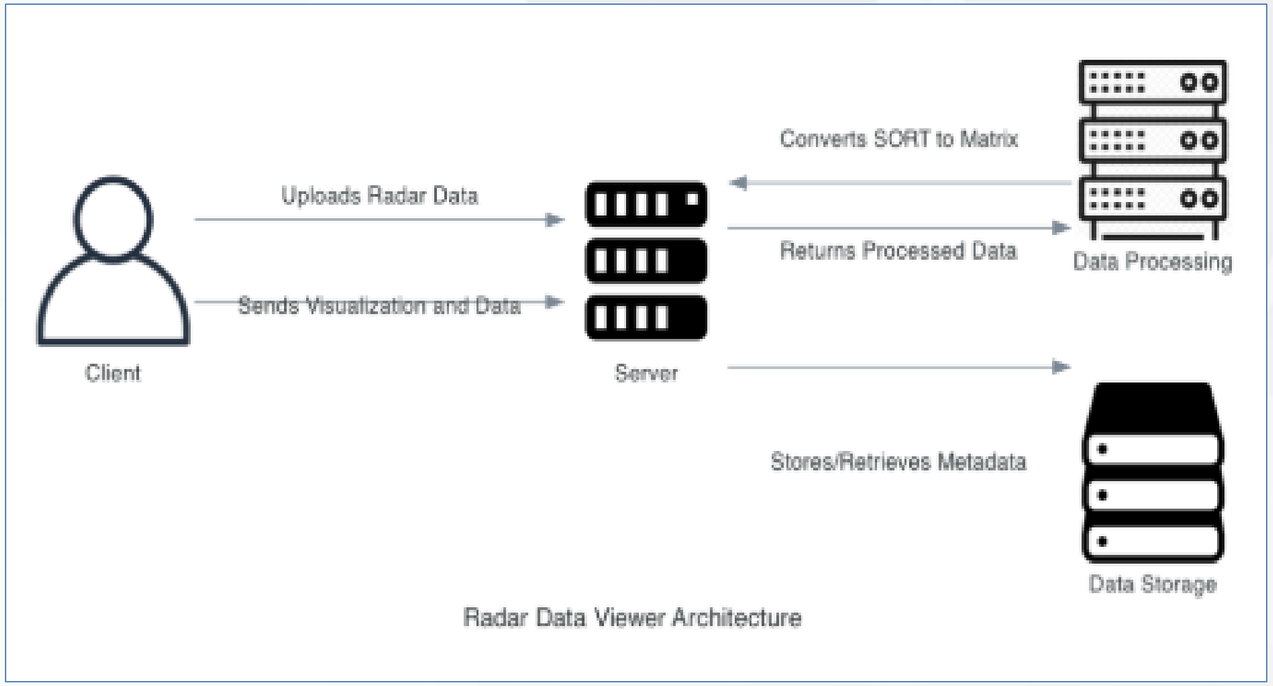
- History of previous data

System Architecture

- **Technologies used:**
  - Backend: Django (Python)
  - Database: SQLite
  - Frontend: React.js
  - Data Processing Libraries: NumPy, SciPy, Pillow, io, base64

System Architecture (Cont.)

- **High-Level Architecture:**
  - Modular client-server model with RESTful APIs.
  - Data processing for radar files using Python libraries.



Radar Data Viewer Architecture

Functional Features

- **File upload Process:**
  - File validation: Accepts only SORT files
  - Metadata extraction and storage
  - Image generation for browser-supported formats (PNG, JPG).
- **Slideshow Feature:**
  - User controls for play, pause, forward, and backward navigation.

Challenges and Mitigations

Problem: Parsing .SORT Files

- Understanding the binary structure, handling inconsistencies, and extracting relevant metadata without predefined schemas.

Mitigation: Analyzed file formats and used numpy and struct for efficient binary parsing.

Problem: Converting Data to Vectors

- Transforming raw radar data into polar vectors and converting to Cartesian coordinates for imaging.

Mitigation: Calculated range and angles, performed polar-to-Cartesian transformations, and normalized intensity.

Problem: Signal Processing

- Reducing noise, applying beamforming, and preparing data for accurate visualization.

Mitigation: Applied Hamming filters and phased array techniques for enhanced data quality.

Challenges and Mitigations (Cont.)

Problem: Creating 3D Matrices:

- Structuring radar data into consistent 3D matrices for high-quality imaging.

Mitigation: 3D Matrices - Organized data into standardized 3D matrices, ensuring consistent image resolution.

Problem: Generating Images:

- Converting processed data into visually interpretable, human-readable images.

Mitigation: Image Generation— Used Pillow to convert 3D matrices into PNG images with normalized pixel values.

Problem: Front-End Display:

- Dynamically rendering radar data while balancing performance and quality.

Mitigation: Front-End Integration— Optimized images for web display and enabled interactive 3D views using tools like WebGL.

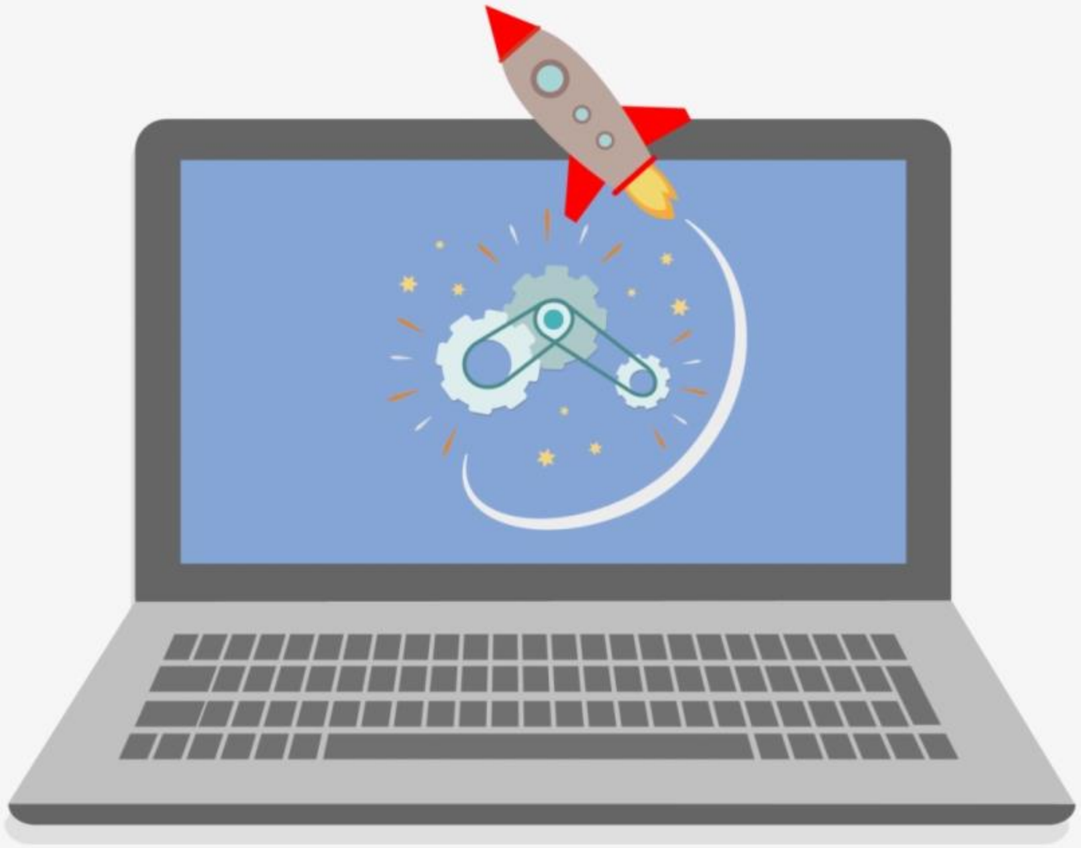
Testing Approach

- **Scope:**
  - Model Testing: File path validation, metadata behavior
  - Utility Testing: Data transformation and image generation
  - User Authentication Testing: Login functionality and edge cases
- **Highlights:**
  - 22+ test cases implemented.
  - Robust error handling for invalid inputs.
  - Ensures smooth functionality across components.

Additional Features

- **CI/CD Integration:**
    - Automated build error checks
    - Runs unit tests during pipeline execution
    - Deployment only after passing all test cases
- **Unit Test Coverage:**
    - Validates individual functions and modules.
    - Ensures system reliability and adherence to quality standards

Demonstration



Conclusion

- **Summary:**
    - Developed an intuitive platform for radar data visualization
    - MVP focuses on B-Scan format and metadata display
    - Future plans include advanced analysis and animations
- **Impact:**
    - Simplifies radar data interpretation for users
    - Sets the foundation for scalable and feature-rich applications

THANK YOU