Pune Vidyarthi Griha's College of Engineering and Technology & G.K. Pate (Wani) Institute of Management,
Pune-411009.
(Affiliated to Savitribai Phule Pune University)

A PRELIMINARY REPORT ON
"Connect Craft"
SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)
Submitted By

STUDENT'S NAME	**EXAM SEAT No.**
ATHARVA OSTWAL	B190074254
YASHSHREE KIRAD	B190074236
ATHARVA PAWAR	B190074264
SUYOG KARPE	B190074234

Under the Guidance of

Prof. B.C.Julme

Department Of Computer Engineering
Academic Year: - 2023-2024

Pune Vidyarthi Griha's College of Engineering and Technology & G.K. Pate (Wani) Institute of Management,
Pune-411009.
(Affiliated to Savitribai Phule Pune University)

I. CERTIFICATE
This is to certify that the project report entitled
"Connect Craft"

STUDENTS'S NAME	**EXAM SEAT No.**
ATHARVA OSTWAL	B190074254
YASHSHREE KIRAD	B190074236
ATHARVA PAWAR	B190074264
SUYOG KARPE	B190074234

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of Prof. B.C.Julme and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Computer Engineering).

Date: -

Place: - Pune

Prof B.C.Julme (Project Guide)

Prof D.D.Sapkal (Head of Computer Department)

(Dr. M. R. Tarambale)
(I/C Principal)
PVG's College of Engineering and Technology and GKPIM, Pune – 09

II. ACKNOWLEDGEMENT

It is an honor to express our gratitude and respect to everyone who has guided and inspired us throughout the project. We would like to thank Prof. B.C.Julme, our Professor-in-charge, for their support and guidance in completing our project 'Connect Craft. It was a great learning experience.

I am also thankful to Prof. D.D. Sapkal, Head of Department, Computer Engineering for the motivation and inspiration and suggestions that ultimately culminated in a favorable outcome.

Atharva Ostwal
Yashshree Kirad
Atharva Pawar
Suyog Karpe

III. ABSTRACT

In our increasingly connected world, network quality is a fundamental concern for users. To assist individuals in making informed decisions about their network service providers, we propose an AI-powered solution. This innovative application harnesses historical network data and real-time user feedback to predict and display potential network quality issues in specific geographic areas.By employing advanced Artificial Intelligence and Machine Learning algorithms, identifying trends and anomalies within this data, allowing it to accurately forecast potential network issues.Key features include predictive analytics, a visual representation of network quality problems on an interactive map,integration of real-time user feedback, provider comparisons based on historical data and reviews,personalized recommendations, and proactive alerts about potential network problems.This comprehensive solution empowers users to select the most suitable network provider for their specific needs, ultimately enhancing their network experiences and satisfaction. It bridges the gap between consumers and network providers by providing transparency and data-driven insights intonetwork quality, ensuring users can confidently make choices that align with their connectivity requirements.

IV. INDEX

V.LIST OF FIGURES

CHAPTER NO 1. INTRODUCTION

1.1.Problem Definition

In the dynamic field of telecommunications, users encounter difficulties in maintaining reliable and consistent network connectivity, especially across diverse geographical landscapes. The demand for precise forecasts of location-based network performance has grown significantly, as users increasingly seek optimal connectivity experiences. This project endeavors to tackle these challenges by introducing an AI-powered system. By harnessing both historical and real-time data, the system aims to predict and elevate location-based network performance. This innovative approach strives to empower users with informed decision-making capabilities when selecting service providers, ensuring that connectivity remains seamless and performance is optimized for the specific demands of different geographical locations. Through this endeavor, the project envisions contributing to an improved and tailored connectivity experience for users navigating through various network challenges.

So our problem statement is to develop an AI powered system -'ConnectCraft' leveraging historical and real-time data for accurate prediction of location based network performance,This solution will empower users to make informed decisions when selecting service providers,ensuring seamless connectivity and optimal performance for their specific geographical needs.

At the heart of "ConnectCraft" lies our unwavering commitment to empowering you, the user. We understand that connectivity is not a one-size-fits-all concept. It's about having the freedom to choose the network provider that seamlessly integrates into your lifestyle and meets your unique demands. Our motivation is simple yet profound: we want to ensure user never tethered to a network that doesn't meet needs. With "ConnectCraft," you're not just searching for a network; you're embarking on a journey toward a more connected, productive, and satisfying digital experience.

1.2.Objectives:

1. **Accurate Network Performance Prediction:** This goal involves developing sophisticated machine learning models that leverage both historical and real-time data to forecast network performance with a high degree of precision. The accuracy of these predictions is vital for empowering users to make well-informed decisions when selecting service providers based on their geographical locations.

2. **Real-time data integration:** It involves establishing seamless connections with external Application Programming Interfaces (APIs) provided by various service providers in the telecommunications industry. The primary goal is to dynamically integrate and update the system with current, real-time network metrics.

3. **User Empowerment:** It encompasses the goal of providing users with the knowledge and tools needed to make informed decisions when selecting service providers, ensuring that their network connectivity aligns with the specific demands of their geographical locations.

4. **Service Provider Accountability :** This objective focuses on creating a transparent framework that holds service providers accountable for the quality of their network services in specific geographical locations. The system aims to empower users with accurate predictions, enabling them to make informed decisions about selecting service providers based on expected network performance.

5. **Feedback Loop:** The feedback loop involves collecting input, insights, and evaluations from users, incorporating these observations into the system's algorithms, and iteratively improving predictive models. This iterative process promotes a dynamic and adaptive system that evolves based on real-world usage and changing network conditions.

6. **Data Privacy and Security:** Ensuring robust data privacy and security is a paramount objective of the AI-Powered Location-Based Network Performance Prediction System. This objective encompasses several key aspects to safeguard user information and uphold ethical standards.

7. **Scalability:** Scalability is a crucial objective in the development of the AI-powered Location-Based Network Performance Prediction System. The goal is to design and implement the system in a way that ensures it can seamlessly handle growth in both user base and data volume, while maintaining optimal performance and responsiveness.

8. **Continuous Improvement:** Continuous improvement is a fundamental objective in the development and operation of the AI-Powered Location-Based Network Performance Prediction System. This objective revolves around an ongoing commitment to refining and enhancing the system over time, ensuring that it remains responsive to changing user needs, technological advancements, and evolving network landscapes.

1.3. Scope:

1. **Data management:** Data management within the scope of the AI-powered location-based network performance prediction system is a crucial aspect that encompasses the collection, storage, processing, and utilization of historical and real-time data. The goal is to ensure the availability, accuracy, and security of data to support the functioning of predictive models and enhance the overall system effectiveness.

2. **Service Providers:** The involvement of service providers within the scope of the AI-powered system is crucial to ensuring that users receive accurate and relevant information when making decisions about network connectivity. The system interacts with and incorporates data from various service providers to enhance the overall predictive capabilities.

3. **Ethical Compliance:** By integrating these ethical considerations into the scope of the project, it ensures that the AI-powered Location-Based Network Performance Prediction System is not only technologically advanced but also aligns with principles of fairness, transparency, and responsible use of predictive analytics.

4. **Geographic Expansion:** The project includes strategies for global reach, adapting the system to diverse regions. It involves localized data partnerships, language features, and cultural sensitivity. Plans encompass entry strategies, legal compliance, and regional partnerships to optimize predictions, ensuring the system's effectiveness across varied geographical landscapes and diverse network infrastructures.

5. **Monitoring and Maintenance:** Incorporate robust monitoring mechanisms to track system performance, ensuring optimal functionality and proactive issue identification. Develop a comprehensive maintenance plan for regular updates, bug fixes, and scalability adjustments, guaranteeing a resilient and continually improving AI-powered Location-Based Network Performance Prediction System.

CHAPTER NO 2. LITERATURE SURVEY

2.1 Real Time Market Analysis

There are many commercial applications which are used by big organisations for analysing network data. These tools are designed considering the organisations as the user. All these software are quite expensive too.

Commercial tools for network analysis:

1. **Nokia AVA AI:** A robust network analytics platform for service providers and large enterprises, optimizing network performance and resource allocation. However, its focus on large-scale networks may limit its cost-effectiveness for smaller organizations.

2. **NetScout nGeniusONE:** Provides real-time network visibility and detailed performance analysis. However, cons include the need for additional hardware, potentially increasing costs and complexity, which may not be suitable for all organizations.

3. **Cisco Prime Infrastructure:** Cisco Prime Infrastructure is a network management tool developed by Cisco Systems. It provides a unified solution for managing wired and wireless networks. As of my last knowledge update in January 2022, I'll provide a general overview of Cisco Prime Infrastructure, but please note that there may have been updates or changes since then.

4. **SolarWinds Network Performance Monitor (NPM):** SolarWinds Network Performance Monitor (NPM) is a comprehensive network management solution providing real-time visibility, customizable dashboards, alerts, and detailed performance analysis. It monitors devices, bandwidth, and applications for optimal network performance in various environments.

5. **Keysight Nemo Outdoor:** Keysight Nemo Outdoor is a leading drive test and benchmarking tool for mobile networks. It provides detailed analysis of network performance, coverage, and quality, supporting technologies such as 5G and delivering insights for optimal network optimization.

6. **PCTEL SeeHawk Touch:** PCTEL SeeHawk Touch is a mobile network testing solution offering real-time analysis for drive testing and in-building mapping. It supports 5G, LTE, and other technologies, providing insights for comprehensive network optimization and troubleshooting.

Thus, by observing and studying the existing systems our brief study came up with new scope and Objectives suitable to today's users and organizations according to their needs.

2.2 Related Research Papers:

1. **Network Strength Comparison by Region:** The research paper, "Network Strength Comparison by Region," delves into network comparisons and security measures. It underscores Reliance Jio's remarkable 20.8 Mbps 4G speed, outperforming competitors, including Airtel. Despite Airtel's faster network, Jio secures a majority of subscriptions, largely attributed to its cost-effective data plans with extended validity. The study reveals the substantial influence of pricing on network preferences in India. This analysis aids in understanding the intricate dynamics of the Indian telecom market, where network speed and affordability are pivotal factors for user choices.

2. **Machine Learning Approaches for Predicting Networkoutages:** Research in this area could explore the application of machine learning algorithms to predict and prevent network outages. The focus might be on developing predictive models for proactive network maintenance.

3. **Security Trends in Broadband ISP Networks: A Global Perspective :** This paper might provide a comprehensive overview of security trends in broadband ISP networks globally, examining emerging threats, countermeasures, and best practices for ensuring network security.

4. **Impact of Networkspeed on User Experience in Emerging Markets:** Research could focus on understanding how variations in network speed affect user experience, particularly in emerging markets. The study might analyze user behavior, satisfaction, and engagement in relation to network speed.

CHAPTER NO 3. SOFTWARE AND HARDWARE REQUIREMENTS

3.1 Software Requirements:

1. **Operating System:** Robust and secure; Linux-based distributions are preferred for server-side components.
2. **Database Management System:** Needed for data storage; options include MySQL, or NoSQL databases like MongoDB.
3. **Programming Languages:** Python (AI and ML), JavaScript/HTML/CSS (UI), Java or C++ (backend).
4. **AI and ML Frameworks:** Utilize TensorFlow, PyTorch, or scikit-learn for model development.
5. **Web Development Frameworks:** Use Django or Flask for UI and RESTful APIs.
6. **Containerization and Orchestration:** Tools for scalable service management.
7. **Data Visualization:** Libraries like Matplotlib, D3.js, or Plotly for presenting data.
8. **Security and Compliance Tools:** Implement security and compliance frameworks for data protection.
9. **Version Control:** Git: For version control and collaboration.
10. **Continuous Integration (CI) / Continuous Deployment (CD):** Jenkins or GitLab CI: For automating testing and deployment processes.

3.2. Hardware Requirements:

1. **Server/Cloud Instances:** Deploying the system on servers or cloud instances with sufficient processing power and memory with Consider cloud services like AWS EC2, Google Cloud Compute Engine, or Azure Virtual Machines for scalability.
2. **CPU:** Multi-core processors (e.g., Intel Xeon or AMD EPYC) for parallel processing, especially beneficial during model training.
3. **RAM:** Adequate RAM to handle large datasets and to support memory-intensive tasks during model training and data processing.
4. **Storage:** Fast and scalable storage, SSDs are preferable for quick data retrieval, especially if dealing with large datasets.
5. **GPUs:** NVIDIA GPUs for ML model training.
6. **Redundancy and Failover:** Ensure high availability and fault tolerance.

7. **Monitoring and Logging Server:** If implementing monitoring and logging systems, allocate resources for these servers to handle the collection and analysis of system logs.

8. **Security Hardware:** Firewalls, intrusion detection, hardware security modules..

9. **Load Balancer:** If deploying the system in a distributed manner, a load balancer can help distribute incoming network traffic across multiple servers, improving performance and reliability.

CHAPTER NO 4. SOFTWARE REQUIREMENTS SPECIFICATION

4.1. Assumptions and Dependencies:

4.1.1. Assumptions:

1. **Data Availability:**
   Assumption: Sufficient historical and real-time data on network performance is available.
   Dependency: Access to reliable and comprehensive datasets from service providers, network infrastructure, and user feedback.

2. **Data Accuracy:**
   Assumption: The provided data is accurate and reflects the true network conditions.
   Dependency: Regular validation and quality checks on the collected data to ensure accuracy.

3. **Relevance of Historical Data:**
   Assumption: Historical data is indicative of future network performance trends.
   Dependency: Monitoring and reassessment of the relevance of historical data over time due to changes in technology, infrastructure, and user behavior.

4. **Stability of Network Infrastructure:**
   Assumption: The network infrastructure remains stable during the prediction period.
   Dependency: Regular updates on the status of network infrastructure and consideration of any planned changes or upgrades.

5. **Consistency in User Behavior:**
   Assumption: User behavior and demands remain consistent over time.
   Dependency: Continuous monitoring of user behavior patterns and adapting the model to changes in usage patterns.

6. **Predictive Model Accuracy**
   Assumption: The chosen AI model accurately predicts location-based network performance.
   Dependency: Regular model validation and refinement based on ongoing feedback and comparison with real-world performance.

7. **Service Provider Cooperation**
   Assumption: Service providers are willing to collaborate and share relevant data for the improvement of the predictive model.
   Dependency: Establishing partnerships and agreements with service providers to access data and enhance the accuracy of predictions.

4.1.2. Dependencies:

1. **Data Integration:**
   Dependency: Integration of diverse data sets from different sources, including service providers, network equipment, and user feedback, into a unified data repository.

2. **Advanced Analytics Tools:**
   Dependency: Availability of advanced analytics tools and platforms for processing large data sets, running complex algorithms, and training machine learning models.

3. **Real-time Data Streaming:**
   Dependency: Implementation of systems for real-time data streaming to continuously update the model with the latest information on network performance.

4. **Machine Learning Expertise:**
   Dependency: Access to expertise in machine learning for developing, training, and maintaining the predictive models.

5. **Geographical Mapping Tools:**
   Dependency: Integration with geographical mapping tools to visualize and interpret location-based network performance predictions effectively.

6. **User Interface (UI) Development:**
   Dependency: Development of a user-friendly interface for users to interact with the predictions and make informed decisions.

7. **Compliance and Privacy:**
   Dependency:Adherence to data protection regulations and obtaining necessary permissions to use and analyze the collected data.

8. **Scalability and Performance:**
   Dependency:Designing the system to be scalable, considering the potential increase in data volume and user interactions.

9. **Feedback Loop:**
   Dependency: Establishment of a feedback loop to gather user feedback and update the model based on real-world experiences.

4.2. Functional Requirements:

4.2.1 Data Ingestion and Collection

1. Historical Data Collector:

1. Requirement 1: Collect historical network performance data from specified sources.
2. Requirement 2: Support various data formats and protocols for ingestion.
3. Requirement 3: Ensure data integrity during the collection process.
4. Requirement 4: Implement a scheduling mechanism for periodic data collection.

2. Real-Time Data Collector:

1. Requirement 5: Continuously gather real-time network performance data.
2. Requirement 6: Handle high-frequency data streams efficiently.
3. Requirement 7: Provide mechanisms for real-time event detection and response.

4.2.2. Data Preprocessing

1. Data Cleaning:

1. Requirement 8: Identify and handle missing or incomplete data points.
2. Requirement 9: Detect and handle outliers or anomalies in the data.
3. Requirement 10: Standardize units and formats for consistency.

2. Data Normalization:

1. Requirement 11: Normalize data to a consistent scale for accurate analysis.
2. Requirement 12: Apply domain-specific normalization techniques.

4.2.3 Data Storage

1. Historical Data Warehouse:

1. Requirement 13: Store historical data in a scalable and efficient data warehouse.
2. Requirement 14: Implement a retrieval mechanism for historical data.
3. Requirement 15: Ensure data security and access controls.

2. Real-Time Data Storage:

1. Requirement 16: Store real-time data with low-latency requirements.
2. Requirement 17: Implement a mechanism for efficient retrieval of real-time data.
3. Requirement 18: Ensure data durability and consistency.

4.2.4 AI/ML Model Development

1. Model Training:

1. Requirement 19: Train machine learning models using historical data.
2. Requirement 20: Implement a mechanism for retraining models periodically.
3. Requirement 21: Support different machine learning algorithms for experimentation.

2. Model Evaluation:

1. Requirement 22: Evaluate model performance using appropriate metrics.
2. Requirement 23: Implement feedback loops for continuous model improvement.
3. Requirement 24: Provide a mechanism to compare and select the best-performing models.

4.2.5 General System Requirements:

1. Scalability:

1. Requirement 25: Design the system to handle increasing data volumes gracefully.

2. Robustness:

1. Requirement 26: Implement error handling and recovery mechanisms.
2. Requirement 27: Ensure the system can handle unexpected data variations.

3. Usability:

1. Requirement 28: Design user interfaces for system monitoring and configuration.
2. Requirement 29: Provide documentation for users and administrators.

4. Security:

1. Requirement 30: Implement encryption for data at rest and during transit.
2. Requirement 31: Enforce access controls to restrict data access based on roles.

4.2.6 Model Deployment:

1. Model :

1. Requirement 32: Deploy trained models as accessible services.
2. Requirement 33: Support versioning of deployed models.
3. Requirement 34: Ensure scalability to handle varying user loads.
4. Requirement 35: Implement monitoring for deployed models to track performance.

4.2.7 User Interface:

1. Location Input Interface:

1. Requirement 36: Provide an intuitive interface for users to input their location.
2. Requirement 37: Validate and sanitize user inputs to ensure accuracy.
3. Requirement 38: Support various location input formats.

2. Prediction Display Interface:

1. Requirement 39: Display network performance predictions in a user-friendly manner.
2. Requirement 40: Update predictions in real-time as new data becomes available.
3. Requirement 41: Enable customization of displayed information based on user preferences.

4.2.8 Recommendation Engine:

1. Recommendation Algorithms:

1. Requirement 42: Develop algorithms that consider both network performance predictions and user preferences.
2. Requirement 43: Allow users to customize preferences for personalized recommendations.
3. Requirement 44: Implement a mechanism to adapt recommendations based on user feedback.

4.2.9 User Feedback Loop:

1. Feedback Collection:

1. Requirement 45: Capture user feedback on recommended service providers and network performance.
2. Requirement 46: Provide multiple feedback channels (e.g., ratings, comments).

2. Feedback Analysis:

1. Requirement 47: Analyze user feedback to improve recommendation algorithms.
2. Requirement 48: Implement sentiment analysis to understand user satisfaction.
3. Requirement 49: Provide insights to administrators for system enhancements.

4.2.10 Monitoring and Maintenance:

1. Monitoring Dashboards:

1. Requirement 50: Display real-time system health metrics, AI model performance, and data source status.
2. Requirement 51: Implement alerting mechanisms for immediate issue identification.
3. Requirement 52: Support customization of dashboard views for different user roles.

4.2.11 Scalability and Performance Optimization:

1. Scalability Framework:

1. Requirement 53: Implement a framework for horizontal scaling of system components.
2. Requirement 54: Monitor system load and automatically scale resources as needed.
3. Requirement 55: Ensure minimal downtime during scaling operations.

2. Performance Optimization Algorithms:

1. Requirement 56: Optimize real-time data processing for low latency.
2. Requirement 57: Continuously refine and enhance algorithms for improved prediction performance.

3. Requirement 58: Implement caching mechanisms for frequently accessed data.

4.2.12 Analytics and Reporting:

1. Reporting Tools:

1. Requirement 59: Allow users to customize and schedule reports.
2. Requirement 60: Ensure export functionality for reports in various formats.

4.2.13 Documentation and Training:

1. Documentation Repository:

1. Requirement 61: Maintain a centralized repository for system architecture documentation.
2. Requirement 62: Provide searchable and indexed documentation for easy retrieval.

4.3. External Interface Requirements:

4.3.1. Software Interface Requirements:

1. **User Interface:**
   1.Intuitive and user-friendly design for easy navigation.
   2.Dashboards displaying real-time network performance metrics.
   3.Interactive maps to visualize location-based performance data.
   4.Filters and search functionality for users to focus on specific regions or service providers.

2. **Data Visualization:**
   1.Graphs and visual representations of network performance metrics.
   2.Heatmaps showing network coverage, congestion, and quality.
   3.Historical performance trends for informed decision-making.

3. **User Authentication and Authorization:**
   1.Secure login mechanism to authenticate users.
   2.Role-based access control to manage user permissions.

4. **Predictive Analytics:**
   1.Clear presentation of predictive analytics results for location-based network performance.
   2.Notifications or alerts for potential issues or recommended actions.

5. **Service Provider Information:**
   1.Detailed profiles of service providers, including performance history.
   2.Comparison tools to assess different service providers based on user-defined criteria.

6. **Feedback Mechanism:**
   1.User feedback forms or mechanisms to collect user experiences.
   2.Integration with customer support or feedback systems for continuous improvement.

7. **Integration with External Systems:**
   1.Integration with external APIs or databases for real-time data updates.
   2.Connectivity with telecom operators' APIs for accessing current network performance metrics.

8. **Map Interactivity:**
   1.Zoom and pan features on interactive maps.
   2.Markers or overlays indicating network performance metrics at specific locations.

9. **Responsive Design:**
   1.Ensure the app is accessible and functional across various devices, including smartphones, tablets, and desktops.

10. **Language and Localization:**
    1.Support for multiple languages to cater to diverse user bases.
    2.Localization features based on user preferences or detected location.

11. **Security Features:**
    1.Encryption of sensitive user data.
    2.Secure transmission of data between the app and the server.

12. **Feedback and Reporting:**
    1.Mechanisms for users to report issues or discrepancies in network performance data.
    2.Reporting tools for administrators to track system usage and performance.

13. **Documentation and Help Section:**
    1.In-app documentation or help sections to guide users on how to interpret performance metrics and use the app effectively.

4.3.2. Hardware Interface Requirements:

1. **Server Infrastructure:**
   1.CPU and RAM: Servers or cloud instances should have sufficient processing power and memory to handle data processing, machine learning model inference, and other computational tasks.
   2.GPU (Optional): Consider using GPUs, especially if the app involves complex machine learning models that can benefit from parallel processing.

2. **Database Server:**
   1.Storage: Database servers should have adequate storage capacity to handle historical data storage and retrieval efficiently.

3. **Network Equipment:**
   1.Routers and Switches: Ensure that network equipment provides reliable and high-speed connectivity, especially for real-time data streaming and communication with external APIs.

4. **Backup and Redundancy Hardware:**
   1.Backup Servers: Set up backup servers or redundant systems to ensure data integrity and application availability in case of hardware failures.

5. **Testing Devices:**
   1.Test Devices: Maintain a diverse set of devices for testing the app's compatibility across various platforms, operating systems, and screen sizes.

6. **Security Hardware:**
   1.Firewalls and Security Appliances: Implement hardware-based firewalls and security appliances to safeguard the application against potential threats.

4.4. Non-Functional Requirements:

4.4.1. Performance Requirements:

1. **Response Time:**
   Requirement: The app should provide responses to user interactions within a specified time frame (e.g., less than 2 seconds for standard queries).
   Justification: Faster response times enhance user experience, especially for real-time data updates and queries.

2. **Real-Time Data Streaming:**
   Requirement: Real-time data updates should be streamed with minimal latency (e.g., less than 1 second delay).
   Justification: Users relying on up-to-date network performance information require timely updates for making informed decisions.

3. **Scalability:**
   Requirement: The app should scale horizontally to handle increasing numbers of users and growing data volumes.
   Justification: Scalability ensures that the app remains responsive and performs well under varying loads.

4. **Concurrency:**
   Requirement: Support a high level of concurrent users accessing the app simultaneously (e.g., thousands of users concurrently).
   Justification: Accommodates usage spikes and ensures a consistent user experience during peak times.

5. **Security Performance:**
   Requirement: Security mechanisms, such as encryption and authentication, should have minimal impact on app performance.
   Justification: Security measures should not compromise the overall speed and responsiveness of the application.

6. **Feedback Submission Time:**
   Requirement: User-submitted feedback and reports should be processed and stored within an acceptable time frame (e.g., less than 5 seconds).
   Justification: Quick processing ensures timely attention to user feedback.

4.4.2. Safety Requirements:

1. **Data Privacy and Protection:**
   Requirement: Implement strong data privacy measures, including encryption of sensitive user data both in transit and at rest.

2. **User Authentication and Authorization:**
   Requirement: Utilize secure user authentication mechanisms, such as two-factor

3. **User Consent and Transparency:**
   Requirement: Obtain clear and informed consent from users regarding data collection, processing, and sharing practices. Provide transparent information about how user data will be used.

4. **Anonymization of Data:**
   Requirement: If sharing aggregated data, ensure that individual user data is anonymized to protect user privacy.

4.4.3. Security Requirements:

1. **Authentication Mechanisms:**
   Requirement: Implement secure and robust authentication mechanisms, including strong password policies and multi-factor authentication.

2. **Authorization Controls:**
   Requirement: Implement role-based access controls to restrict user access to specific functionalities based on their roles and responsibilities.

3. **Intrusion Detection and Prevention:**
   Requirement: Implement intrusion detection and prevention mechanisms to identify and respond to potential security threats.

4. **Data Backup and Recovery:**
   Requirement: Implement regular data backups and establish a robust data recovery plan to minimize data loss in the event of a security incident.

4.4.4. Software Quality Requirements:

1. **Compatibility:**
   Requirement: he app should be compatible with a range of devices, browsers, and operating systems.

2. **Maintainability:**
   Requirement: The app's code base should be well-organized and documented, making it easy for developers to understand and maintain.

3. **Documentation:**
   Requirement: Provide comprehensive documentation for users, administrators, and developers, including user guides, API documentation, and system architecture documentation.

4. **Error Handling:**
   Requirement: Implement robust error handling mechanisms to provide meaningful error messages and log errors for analysis.

5. **User Training and Support:**
   Requirement: Provide user training materials and support channels to assist users in understanding and effectively using the app.

4.5. System Requirements:

4.5.1. Database Requirements:

1. **Data Security:** Implement robust access controls and encryption measures to secure sensitive user and network performance data.
2. **Indexing:** Implement appropriate indexing strategies to optimize query performance, especially for frequently accessed data.
3. **Real-Time Data Storage:** Provide mechanisms to store and update real-time network performance data for immediate access and analysis.
4. **Database Monitoring:** Implement monitoring tools to track database performance, identify potential issues, and ensure optimal operation.
5. **Database Compliance:** Ensure that the database design and operations comply with relevant data protection regulations and industry standards.
6. **Data Governance:** Implement data governance practices, including data ownership, data quality management, and data access policies.
7. **Database Performance Testing:** Conduct performance testing of database operations to ensure that the database can handle expected loads and respond within acceptable time frames.

4.5.2. Software Requirements:

1. **Operating System:** Robust and secure; Linux-based distributions are preferred for server-side components.
2. **Database Management System:** Needed for data storage; options include MySQL, or NoSQL databases like MongoDB.
3. **Programming Languages:** Python (AI and ML), JavaScript/HTML/CSS (UI), Java or C++ (backend).
4. **AI and ML Frameworks:** Utilize TensorFlow, PyTorch, or scikit-learn for model development.
5. **Web Development Frameworks:** Use Django or Flask for UI and RESTful APIs.
6. **Containerization and Orchestration:** Tools for scalable service management.
7. **Data Visualization:** Libraries like Matplotlib, D3.js, or Plotly for presenting data.
8. **Security and Compliance Tools:** Implement security and compliance frameworks for data protection.
9. **Version Control:** Git: For version control and collaboration.

10. **Continuous Integration (CI) / Continuous Deployment (CD):** Jenkins or GitLab CI: For automating testing and deployment processes.

4.5.3. Hardware Requirements:

1. **Server/Cloud Instances:** Deploying the system on servers or cloud instances with sufficient processing power and memory with Consider cloud services like AWS EC2, Google Cloud Compute Engine, or Azure Virtual Machines for scalability.
2. **CPU:** Multi-core processors (e.g., Intel Xeon or AMD EPYC) for parallel processing, especially beneficial during model training.
3. **RAM:** Adequate RAM to handle large datasets and to support memory-intensive tasks during model training and data processing.
4. **Storage:** Fast and scalable storage, SSDs are preferable for quick data retrieval, especially if dealing with large datasets.
5. **GPUs:** NVIDIA GPUs for ML model training.
6. **Redundancy and Failover:** Ensure high availability and fault tolerance.
7. **Monitoring and Logging Server:** If implementing monitoring and logging systems, allocate resources for these servers to handle the collection and analysis of system logs.
8. **Security Hardware:** Firewalls, intrusion detection, hardware security modules..
9. **Load Balancer:** If deploying the system in a distributed manner, a load balancer can help distribute incoming network traffic across multiple servers, improving performance and reliability.

4.6. Analysis Model: SDLC Model

4.6.1. Planning:

1. **Objective:** Define the goals and scope of the network performance analysis system.
2. **Activities:**
   (a) Conduct a thorough analysis of business and user requirements.
   (b) Define project milestones, timelines, and resource requirements.
   (c) Establish the feasibility of the project, considering technical, economic, and organizational aspects.

4.6.2. System Design:

1. **Objective:** Develop a detailed blueprint of the entire system architecture.
2. **Activities:**
   (a) Design the overall system architecture, including data flow diagrams and component interactions.
   (b) Specify the design of data ingestion components, data preprocessing modules, storage structures, AI/ML model development, and deployment components.
   (c) Design user interfaces for location input and prediction display.
   (d) Define the architecture and interfaces for external integrations (e.g., provider information databases).

4.6.3. Implementation:

1. **Objective:** Transform the design into a working system.
2. **Activities:**
   (a) Develop data collectors for historical and real-time data.
   (b) Implement data preprocessing components for cleaning and normalization.
   (c) Build storage mechanisms for historical and real-time data.
   (d) Develop AI/ML model training and evaluation modules.
   (e) Implement model deployment as APIs.
   (f) Create user interfaces for location input and prediction display.
   (g) Integrate provider information databases and recommendation algorithms.

4.6.4. Testing:

1. **Objective:** Verify and validate the functionality and performance of the system.

2. **Activities:**
   (a) Conduct unit testing for individual components.
   (b) Perform integration testing to ensure seamless communication between different modules.
   (c) Execute system testing to validate end-to-end functionality.
   (d) Carry out performance testing to ensure scalability and responsiveness.
   (e) Implement user acceptance testing (UAT) with stakeholders.

4.6.5. Deployment:

1. **Objective:** Release the system for production use.
2. **Activities:**
   (a) Deploy the system in a controlled environment.
   (b) Ensure data migration and integrity during deployment.
   (c) Monitor the system closely during the initial deployment phase.
   (d) Provide necessary training for administrators and end-users.

4.6.6. Monitoring and Maintenance:

1. **Objective:** Ensure the ongoing health and performance of the system.
2. **Activities:**
   (a) Set up monitoring dashboards for system health, AI model performance, and data sources.
   (b) Implement automated alerts for issue identification.

4.6.7. Documentation and Training:

1. **Objective:** Provide comprehensive documentation and training materials.
2. **Activities:**
   (a) Develop and maintain a centralized repository for system architecture documentation.
   (b) Regularly update documentation to reflect system changes.

4.6.8. Review and Continuous Improvement:

1. **Objective:** Continuously enhance the system based on feedback and changing requirements.
2. **Activities:**

(a) Collect and analyze user feedback on network performance and recommendations.

(b) Use feedback to improve AI/ML models and recommendation algorithms.

SDLC Model Selection:
Given the iterative and evolving nature of the system, an Agile SDLC model may be well-suited. Agile allows for flexibility, adaptability to changing requirements, and incremental development cycles. Regular sprint reviews and retrospectives can facilitate continuous improvement, and feedback loops can be integrated at various stages to ensure user satisfaction and system effectiveness.

CHAPTER NO 5. SYSTEM DESIGN

5.1 System Architecture:

The system architecture is organized into two main subgraphs: UserInterface and System. The UserInterface subgraph represents the user-facing components, while the System subgraph comprises the backend components responsible for data collection, processing, modeling, and maintenance.

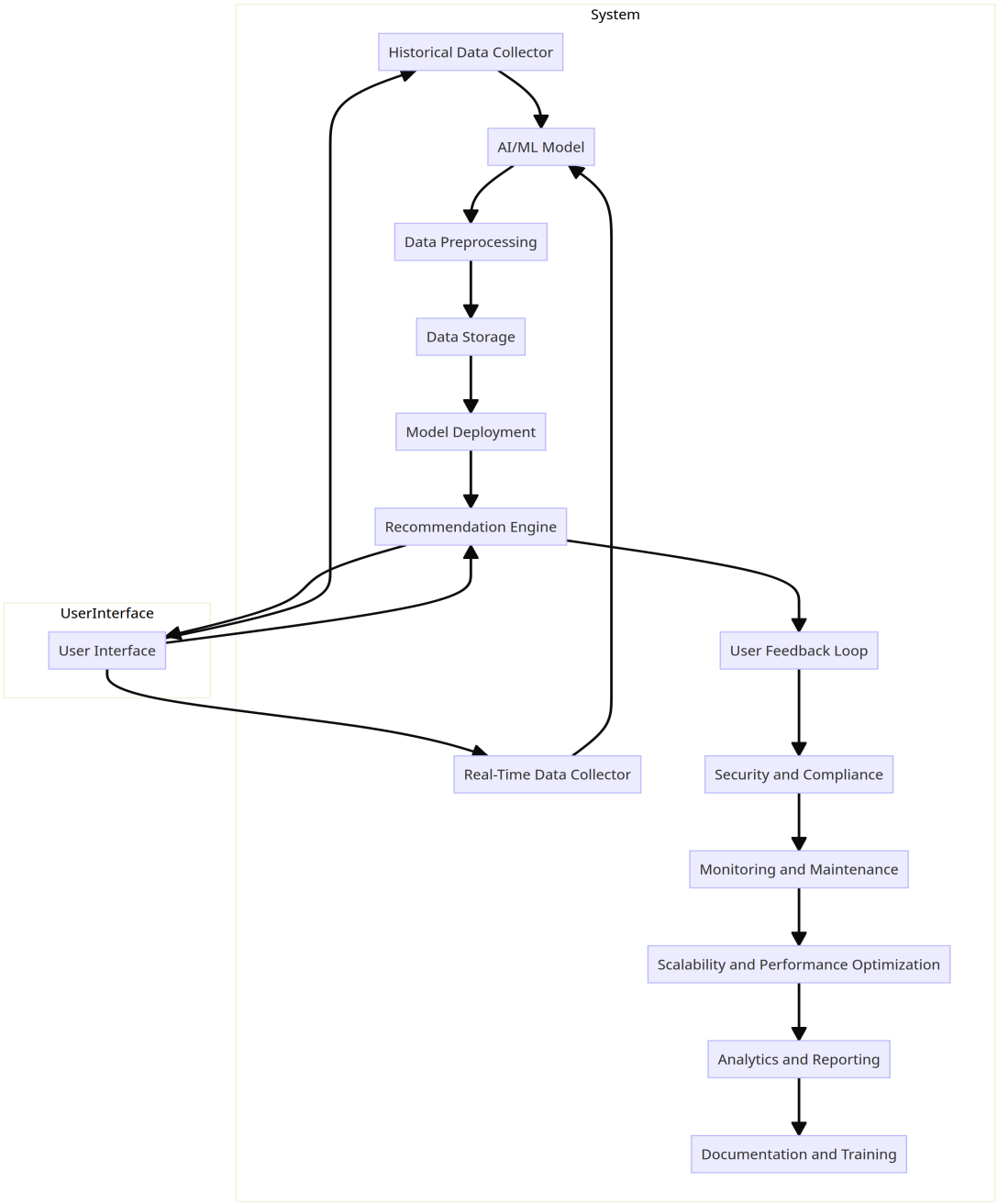

Fig. 5.1 System Architecture

1. User Interface:

1. **A [User Interface]:**
    - **Description:** This component represents the user interface, providing users with the means to interact with the system. It includes interfaces for location input, prediction display, and feedback submission.

2. System:

1. **B [Historical Data Collector]:**
    - **Description:** The Historical Data Collector is responsible for gathering historical network performance data from specified sources.
2. **C [Real-Time Data Collector]:**
    - **Description:** The Real-Time Data Collector continuously collects real-time network performance data.
3. **D [AI/ML Model]:**
    - **Description:** The AI/ML Model component is responsible for developing and fine-tuning machine learning models for network performance prediction.
4. **E [Data Preprocessing]:**
    - **Description:** Data Preprocessing encompasses the cleaning, normalization, and preparation of collected data for analysis.
5. **F [Data Storage]:**
    - **Description:** The Data Storage component stores preprocessed historical and real-time data for retrieval.
6. **G [Model Deployment]:**
    - **Description:** Model Deployment involves deploying trained models as APIs or services for user access.
7. **H [Recommendation Engine]:**
    - **Description:** The Recommendation Engine suggests service providers based on predictions and user preferences.
8. **I [User Feedback Loop]:**
    - **Description:** The User Feedback Loop captures and analyzes user feedback on recommended providers and network performance.
9. **J [Security and Compliance]:**

- **Description:** This component ensures the security and compliance of the system, implementing measures to protect user data.

10. **K [Monitoring and Maintenance]:**

    - **Description:** Monitoring and Maintenance components oversee system health, AI model performance, and data sources. Regular updates to AI models are scheduled.

11. **L [Scalability and Performance Optimization]:**

    - **Description:** This component ensures the system can scale horizontally and optimizes real-time data processing and predictions.

12. **M [Analytics and Reporting]:**

    - **Description:** Analytics and Reporting components provide users and administrators with insights into network performance. Reporting tools and an analytics engine contribute to this functionality.

13. **N [Documentation and Training]:**

    - **Description:** The Documentation and Training component includes materials to educate administrators and users about the system. A documentation repository maintains system architecture and processes.

5.2 List of Modules

1. **Data Ingestion and Collection**
    (a) *Historical and Real-Time Data Collectors*
2. **Data Preprocessing**
    (a) *Data Cleaning and Normalization*
    (b) *Data Storage*
        i. *Historical Data Warehouse*
        ii. *Real-Time Data Storage*
3. **AI/ML Model Development**
    (a) *Model Training and Evaluation*
    (b) *Model Deployment*
4. **User Interface**
    (a) *Location Input Interface*
    (b) *Prediction Display Interface*
5. **Provider Information Integration**
    (a) *Provider Information Database/API*
6. **Recommendation Engine**
    (a) *Recommendation Algorithms*
7. **User Feedback Loop**
    (a) *Feedback Collection and Analysis*
8. **Monitoring and Maintenance**
    (a) *Monitoring Dashboards*
    (b) *Model Update Scheduler*
9. **Scalability and Performance Optimization**
    (a) *Scalability Framework*
    (b) *Performance Optimization Algorithms*
10. **Analytics and Reporting**
11. **Documentation and Training**

5.3 Functionality

1. Data Ingestion and Collection:

Functionality: Gathers historical and real-time network performance data for analysis.

Components:

1.1 Historical Data Collector: Responsible for collecting historical network performance data.

1.2 Real-Time Data Collector: Gathers real-time network performance data for immediate analysis.

2. Data Preprocessing:

Functionality: Cleans, normalizes, and prepares collected data for analysis.

Components:

2.1 Data Cleaning: Ensures data integrity by cleaning and removing inconsistencies.

2.2 Normalization Module: Normalizes data to a consistent format for uniform analysis.

3. Data Storage:

Functionality: Stores preprocessed historical and real-time data for retrieval.

Components:

3.1 Historical Data Warehouse: Storage for processed historical network performance data.

3.2 Real-Time Data Storage: Houses preprocessed real-time network performance data.

4. AI/ML Model Development:

Functionality: Develops and fine-tunes machine learning models for network performance prediction.

Components:

4.1 Model Training Module: Trains machine learning models using historical and real-time data.

4.2 Evaluation Module: Assesses the performance of trained models.

5. Model Deployment:

Functionality: Deploys trained models as APIs or services for user access.

Components:

5.1 Model APIs: Exposes trained models as APIs for seamless user interaction.

6. User Interface:

Functionality: Provides user input interfaces for location selection and displays network performance predictions.

Components:

6.1 Location Input Interface: Enables users to input location and preferences.

6.2 Prediction Display Interface: Displays network performance predictions to users.

7. Provider Information Integration:

Functionality: Integrates information about service providers, including coverage areas, pricing, and quality data.

Components:

7.1 Provider Information Database/API: Centralized repository/API for service provider details.

8. Recommendation Engine:

Functionality: Suggests service providers based on predictions and user preferences.

Components:

8.1 Recommendation Algorithms: Utilizes algorithms to suggest optimal service providers.

9. User Feedback Loop:

Functionality: Captures and analyzes user feedback on recommended providers and network performance.

Components:

9.1 Feedback Collection: Gathers user feedback on recommended providers.

9.2 Feedback Analysis: Analyzes user feedback to enhance system recommendations.

10. Monitoring and Maintenance:

Functionality: Monitors system health, AI model performance, and data sources. Regularly updates AI models.

Components:

10.1 Monitoring Dashboards: Visualizes system health, AI model performance, and data sources.

10.2 Model Update Scheduler: Schedules and manages regular updates for AI models.

11. Scalability and Performance Optimization:

Functionality: Ensures the system can scale horizontally and optimizes real-time data processing and predictions.

Components:

11.1 Scalability Framework: Facilitates horizontal scaling for increased system capacity.

11.2 Performance Optimization Algorithms: Enhances real-time data processing and prediction efficiency.

12. Analytics and Reporting:

Functionality: Provides analytics and reporting capabilities for users and administrators.

Components:

12.1 Reporting Tools: Generates comprehensive reports for users and administrators.

12.2 Analytics Engine: Performs in-depth analysis to extract meaningful insights.

13. Documentation and Training:

Functionality: Documents system architecture and processes, and provides training materials for administrators and users.

Components:

13.1 Documentation Repository: Centralized storage for system architecture documentation.

5.4 UML Diagrams

5.4.1 Sequence Diagram:

The sequence diagram provides a comprehensive overview of interactions within the network performance analysis system, illustrating the flow of actions from user input to continuous system improvement based on feedback. The key participants include the User, initiating interaction by entering location and preferences, and the User Interface, forwarding inputs to the System. The System encompasses various components such as HistoricalDataCollector and RealTimeDataCollector for data collection, AIModel for machine learning model development, DataPreprocessing for data cleaning, DataStorage for storage, and ModelDeployment for deploying trained models. The RecommendationEngine suggests service providers based on predictions and user preferences. The UserFeedbackLoop captures and analyzes user feedback on recommendations, contributing to ongoing system enhancement.

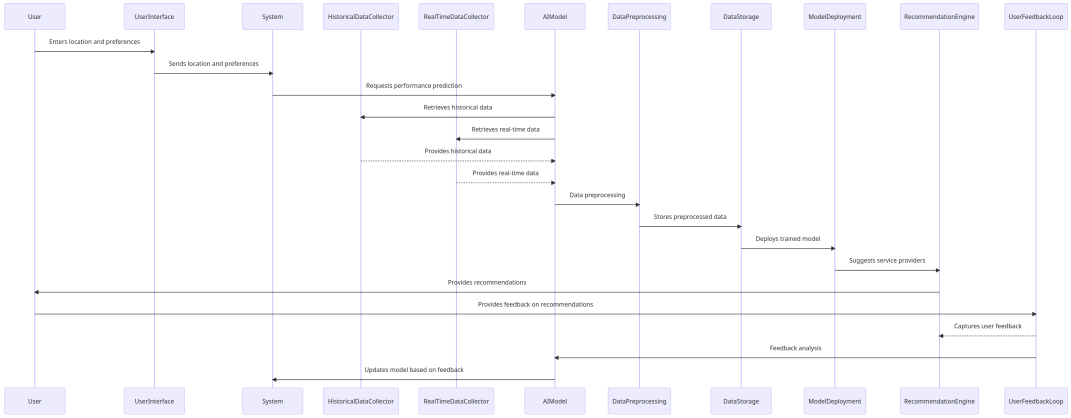

Fig. 5.4.1 Sequence Diagram

5.4.2 Data Flow Diagram:

The Level 0 Data Flow Diagram (DFD) presents a top-level depiction of the network performance analysis system, outlining key interactions and data flows. The User Subsystem involves the User Interface (A) facilitating user interactions and nthe User (B) providing input and receiving performance forecasts. The System Subsystem comprises components such as Historical Data Collector (C) and Real-Time Data Collector (D) for data collection, AI/ML Model (E) for performance prediction, Data Preprocessing (F) for data preparation, Data Storage (G) for storage, Model Deployment (H) for deploying trained models, Recommendation Engine (I) for service suggestions, and User Feedback Loop (J) for feedback analysis. The data flow begins with user input, followed by data collection, modeling, preprocessing, storage, model deployment, and recommendations. The User Feedback Loop captures and analyzes user feedback on recommendations. This Level 0 DFD succinctly illustrates the major components and data interactions, emphasizing the system's overall flow.

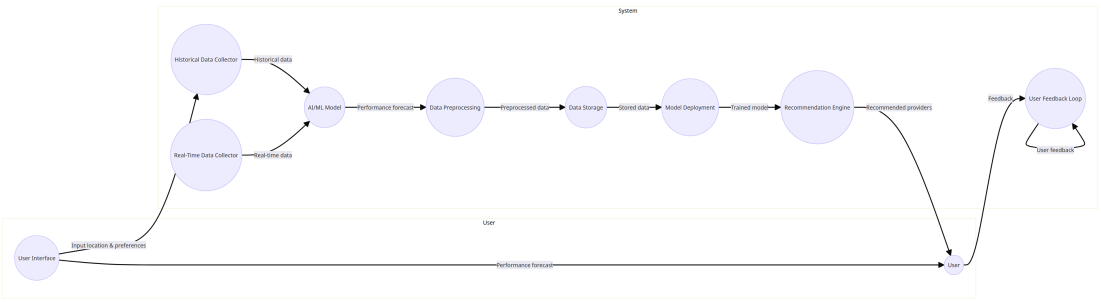

Fig. 5.4.2 Data Flow Diagram

5.4.3 Class Diagram:

The class diagram outlines the architecture of a network performance analysis system, featuring essential classes with specific attributes and functionalities. The ”User” class, including UserID, Name, and Email, represents individuals interacting with the system and has a ”Uses” relationship with the ”ServiceProvider” class, embodying ProviderID, Name, and LocationID.
Temporal aspects are captured by the ”HistoricalData” and ”RealTimeData” classes, with DataID, LocationID, DataValue, and Timestamp attributes. Users are linked to historical and real-time data through ”UsesHistoricalData” and ”UsesRealTimeData” relationships. The ”AIModel” class, featuring ModelID and LocationID, plays a crucial role in performance prediction and establishes relationships (”RelatesTo”) with historical and real-time data.

The ”Location” class, with LocationID and Name attributes, provides contextual information, and the ”Associated With” relationship signifies the association between an AIModel and a specific location. This detailed class diagram encapsulates entities, attributes, and relationships, serving as a visual blueprint for system design and fostering understanding among stakeholders. It facilitates the development of a robust network performance analysis framework.

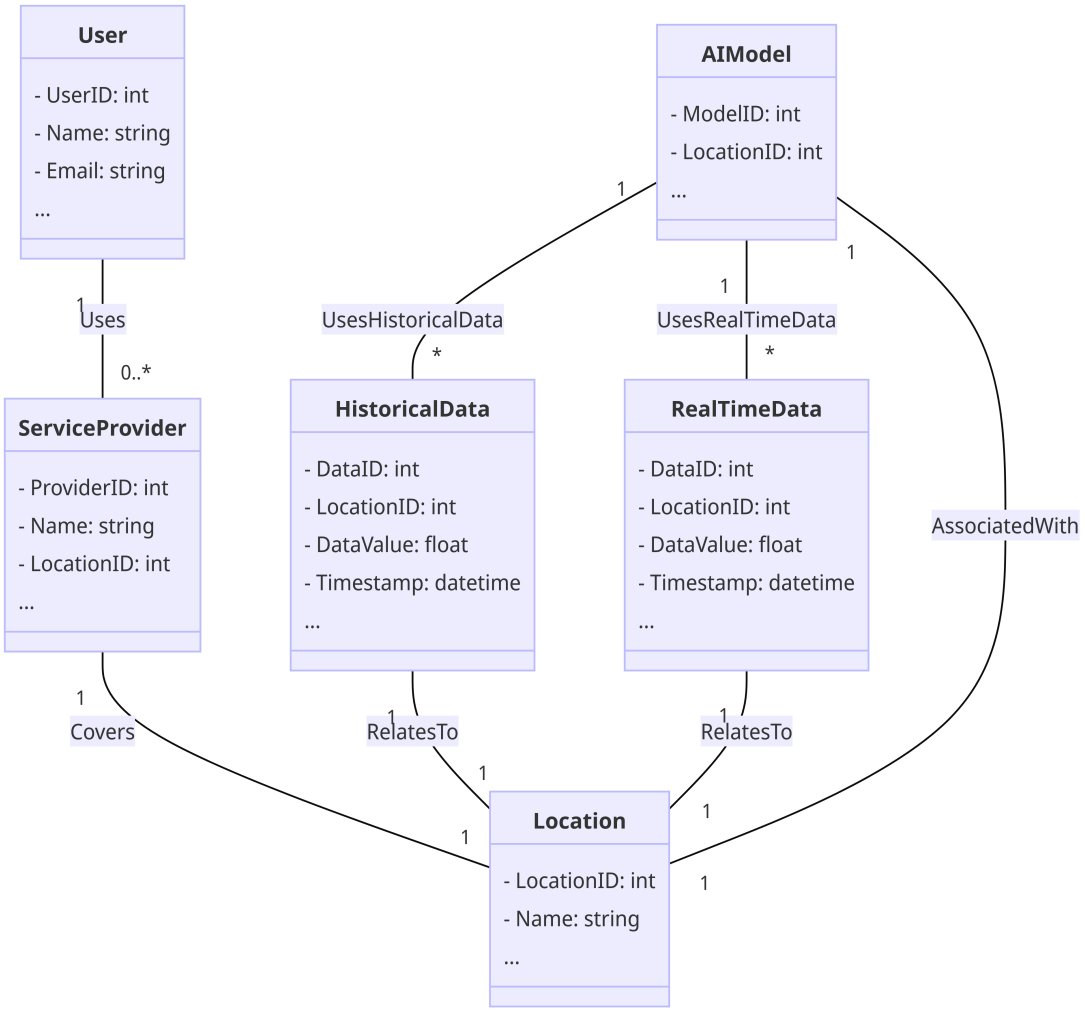

Fig. 5.4.3 Class Diagram

5.4.4 ER Diagram:

Entities in the ER diagram represent key classes in the network performance analysis system. The ”User” entity, uniquely identified by UserID, encapsulates attributes like Name and Email. Similarly, the ”ServiceProvider” entity corresponds to the ”ServiceProvider” class, featuring attributes ProviderID, Name, and LocationID. The ”HistoricalData” and ”RealTimeData” entities capture attributes DataID, LocationID, DataValue, and Timestamp. The ”AIModel” entity includes attributes ModelID and LocationID, while the ”Location” entity encompasses LocationID and Name.

The relationships in the ER diagram mirror those in the class diagram. The ”Uses” relationship between the ”User” and ”ServiceProvider” entities signifies that a user can utilize services from multiple providers. The ”UsesHistoricalData” and ”UsesRealTimeData” relationships with ”User” denote the utilization of historical and real-time data, respectively. The ”RelatesTo” relationship between ”AIModel” and both ”HistoricalData” and ”RealTimeData” entities signifies the association between an AI model and historical/real-time data. Lastly, the ”Associated With” relationship between ”AIModel” and ”Location” represents the connection between an AI model and a specific location in the system.

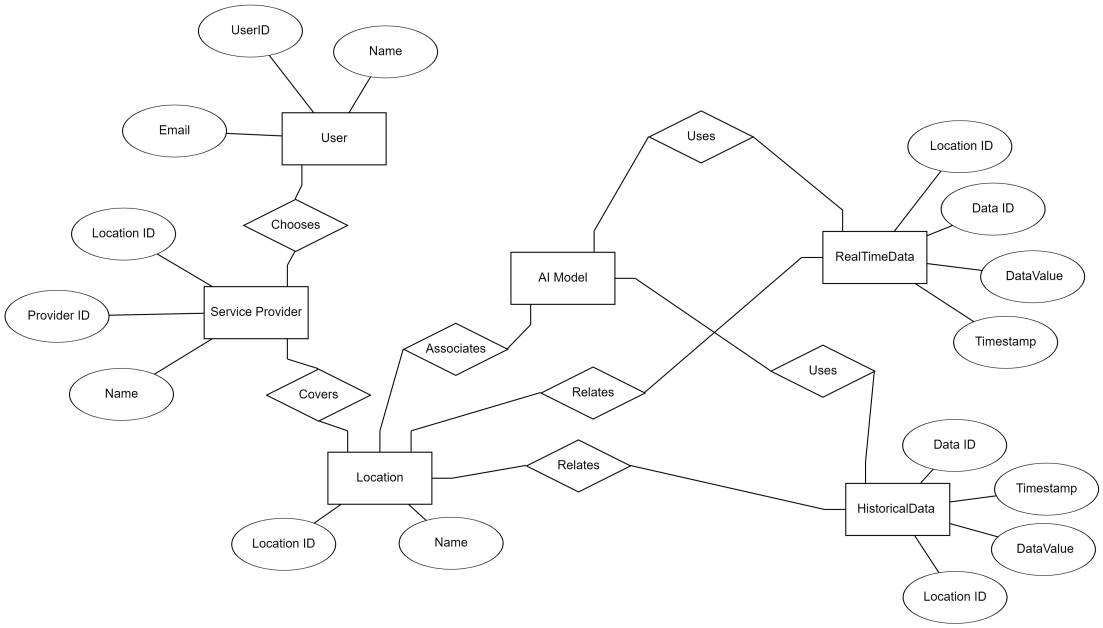

Fig. 5.4.4 ER Diagram

CHAPTER NO 6. PROJECT PLAN

"ConnectCraft" is driven by a steadfast commitment to empowering users through a personalized and seamless network experience. Focused on accurate network performance prediction, real-time data integration, and historical data utilization, the project aims to enhance user empowerment, transparency, and decision-making. The ultimate goal is to deliver an enriched user experience with a strong emphasis on choice, provider selection, accountability, feedback loops, data privacy, security, scalability, and continuous improvement.

Project Phases:

Initiation: In the initiation phase, we define the project scope, objectives, and key deliverables. Establishing a dedicated project team, assigning roles, and conducting a kickoff meeting ensures alignment among stakeholders.

Planning: Detailed project planning, including timelines, milestones, risk analysis, and resource requirements, is a critical step in setting the foundation for the successful execution of "ConnectCraft."

Design: The design phase involves outlining the architecture for accurate network performance prediction, designing real-time data integration and historical data utilization modules, and developing user-centric features that prioritize empowerment and informed decision-making.

Development: Implementation of the designed modules, integration of real-time data sources, and optimization of historical data utilization are the key focus areas during the development phase. Building user interfaces that enhance user experience and choice is also paramount.

Testing: Thorough testing of network performance prediction algorithms, validation of real-time data integration and historical data utilization, and user acceptance testing with feedback collection are integral to ensuring the robustness of the "ConnectCraft" platform.

Deployment: During deployment, the platform is released in a controlled environment. Attention is given to ensuring smooth data migration, system integrity, and close monitoring during the initial deployment phase.

Monitoring and Optimization: Setting up monitoring dashboards for system health and performance, implementing automated alerts, and optimizing network performance and scalability based on monitoring results are essential aspects of this phase.

Documentation and Training: Developing and maintaining comprehensive documentation for system architecture, creating training materials for administrators and end-users, and conducting training sessions contribute to effective system utilization.

Review and Continuous Improvement: Collecting and analyzing user feedback, utilizing this feedback to enhance prediction algorithms and user interfaces, and regularly reviewing and updating system components ensure continuous improvement and alignment with evolving needs.

CHAPTER NO 7.EXPECTED OUTCOME

The envisioned outcome of the "ConnectCraft" project is a revolutionary shift in user connectivity, marked by several key deliverables. Firstly, the development and implementation of precise algorithms for network performance prediction aim to provide users with accurate insights into the performance of different network providers, fostering more informed decision-making. The seamless integration of real-time data sources and effective utilization of historical data are expected to enhance the accuracy and relevance of predictions, ensuring a dynamic and up-to-date perspective.

"ConnectCraft" seeks to empower users with features that put them at the center of their connectivity experience. Through user-centric interfaces, the project aims to provide transparency in network performance metrics, enabling users to make choices aligned with their preferences. The platform aspires to introduce a sense of control and personalization, fostering an environment where users can tailor their network selection to meet their unique demands.

Additionally, the project emphasizes accountability in the network ecosystem. By introducing mechanisms that hold service providers accountable for the quality and reliability of their services, "ConnectCraft" envisions a more competitive and responsible landscape. The implementation of a robust feedback loop will capture and analyze user input, driving continuous improvement in network performance predictions and user interfaces.

Furthermore, "ConnectCraft" prioritizes data privacy and security, implementing measures to ensure the confidentiality of user data. The project's scalable system architecture is designed to accommodate the evolving demands of an ever-changing network landscape, ensuring adaptability and longevity. The continuous cycle of improvement, guided by user feedback and technological advancements, will keep the platform agile and responsive.

Ultimately, the expected outcome is an elevated user experience, marked by satisfaction, engagement, and a newfound sense of choice and control in the digital connectivity realm. The successful implementation of comprehensive documentation and training materials will further contribute to the effectiveness and widespread adoption of the "ConnectCraft" platform by administrators and end-users alike. In summary, "ConnectCraft" aims to redefine the connectivity landscape, offering users a connected, productive, and satisfying digital experience.

CONCLUSION

In conclusion, we have delved into the intricate dynamics of network strength comparison and security in the context of modern telecommunications. Our findings underscored the significance of affordability, reliability, and speed as primary factors influencing user preferences in network selection. Furthermore, the crucial role of robust security measures and data compliance in safeguarding user data and network integrity cannot be overstated. As we navigate an ever-evolving landscape of network technologies, it contributes to a deeper understanding of network performance and user behavior, offering practical recommendations for service providers, users, and policymakers alike.

Probable Date of Completion

We estimate a probable date of completion around February 28, 2024.

REFERENCES

1. Sakthi S M,Kishore Ananth N,Roshini R S,Thenisha S,Dharani kumar "Network Strength Comparison by Region," International Journal of Trendy Research in Engineering and Technology Volume 6 Issue 3 June 2022

2. Xiaohong Deng, Yun Feng, Thanchanok Sutjarittham, Hassan Habibi Gharakheili, Blanca Gallego, Vijay Sivaraman "Comparing Broadband ISP Performance using Big Data from M-Lab "

3. Emmanuel Basikolo, Thomas Basikolo "TOWARDS ZERO DOWNTIME: USING MACHINE LEARNING TO PREDICT NETWORK FAILURE IN 5G AND BEYOND", International Telecommunication Union, 2023.

4. Dr Dipankar Dey,Solanki Pattanayak "Security Trends in Broadband ISP Networks: A Global Perspective "

Appendix A

The project aims to develop an AI-driven solution for predicting location-based network performance, facilitating informed provider selection. Leveraging historical and real-time data is crucial for accurate predictions. The feasibility assessment evaluates data availability and computational resources, ensuring the practicality of the proposed solution. The satisfiability analysis examines the existence of a solution meeting the constraints. Computational complexity is classified to understand the problem's level of difficulty. Modern algebra and mathematical models, including relevant algebraic structures, are applied for data representation. The proposed solutions involve machine learning models and predictive analytics, aiming to offer a ranking system for provider selection. Validation includes experiments with historical and real-time data, assessing prediction accuracy and decision-making effectiveness. Comprehensive documentation covers the methodology, model details, and overall project structure. This structured approach provides clarity in addressing the problem, feasibility, and proposed solutions for predicting location-based network performance.

Appendix B

1. Sakthi S M, Kishore Ananth N, Roshini R S, Thenisha S, Dharani Kumar A B - "Network Strength Comparison by Region" explores the comparison of network strength across different regions.

2. Xiaohong Deng, Yun Feng, Thanchanok Sutjarittham, Hassan Habibi Gharakheili, Blanca Gallego, Vijay Sivaraman - In "Comparing Broadband ISP Performance using Big Data from M-Lab," the authors delve into the comparison of broadband ISP performance using big data from M-Lab.

3. Emmanuel Basikolo, Thomas Basikolo - "Towards Zero Downtime: Using Machine Learning to Predict Network Failure in 5G and Beyond" discusses the application of machine learning to predict network failure, particularly in the context of 5G technology.

4. Dr Dipankar Dey, Solanki Pattanayak - "Security Trends in Broadband ISP Networks: A Global Perspective" provides a global perspective on security trends in broadband ISP networks.

Appendix C

Page Count: 41
Word Count: 7173
Unique: 90.2%
Plagiarism: 9.8%