ISRO Satellite Integration and Test Establishment (ISITE)

DATABASE USER INTERFACE DESIGN FOR SPACECRAFT TELEMETRY & TELECOMMAND SYSTEM RF CHARACTERIZATION SOFTWARE

PROJECT REPORT

Submitted by,

DHESIKA S

In fulfillment of the

INTERNSHIP

in

SPACECRAFT CHECKOUT GROUP

ISRO Satellite Integration and Test Establishment (ISITE),

U. R. RAO SATELLITE CENTRE (Formerly known as ISRO Satellite Centre (ISAC)),

Department of Space, Government of India.

Bengaluru, India.

NOV- DEC 2023

ISRO Satellite Integration and Test Establishment (ISITE)

BONAFIDE CERTIFICATE

Certified that this project report "**DATABASE USER INTERFACE DESIGN FOR SPACECRAFT TELEMETRY & TELECOMMAND SYSTEM RF CHARACTERIZATION SOFTWARE**" is the bonafide work of Dhesika S, who learned and worked on some components of the project undertaken at **'SPACECRAFT CHECKOUT GROUP'.**

GUIDE	DIVISION HEAD
Mr. Janardhan Reddy, Scientist/Engg-SF, Spacecraft Checkout Group, ISRO Satellite Integration and Test Establishment (ISITE), Department of Space, Government of India, Bengaluru.	**Smt. Bharathi,** Division Head, Spacecraft Checkout Group, ISRO Satellite Integration and Test Establishment (ISITE), Department of Space, Government of India, Bengaluru.

Submitted for the fulfillment of the internship held during Nov – Dec 2023.

DECLARATION

I, hereby jointly declare that the project work entitled **"DATABASE USER INTERFACE DESIGN FOR SPACECRAFT TELEMETRY & TELECOMMAND SYSTEM RF CHARACTERIZATION SOFTWARE",** submitted in fulfillment of the '**Internship'**, is a report of the **DHESIKA S** and the project done at '**SPACECRAFT CHECKOUT GROUP**' under the guidance of **Mr.Janardhan,** Scientist/Engg-SF, Spacecraft Checkout Group, ISRO Satellite Integration and Test Establishment (ISITE), Department of Space, Bengaluru, India.

NAME	SIGNATURE
DHESIKA S	

I certify that the declarations made by the above candidate is true.

GUIDE

Mr. Janardhan Reddy,
Scientist/Engg-SF,
Spacecraft Checkout Group,
ISRO Satellite Integration and Test
Establishment (ISITE), URSC / ISRO,
Department of Space, Government of India.

ACKNOWLEDGMENT

I take this opportunity to extend our heartfelt thanks and deep respect to all those who have guided and inspired me throughout the journey of this internship. My journey at the **U. R. RAO SATELLITE CENTRE** (formerly known as **ISRO** Satellite Centre (ISAC)), under the aegis of the Department of Space, Government of India, has been an enriching experience. The provision of wonderful infrastructure and a conducive learning environment has been instrumental in our professional growth.

I am especially grateful to the Group Director, **Mr. Krishnan** and Division Head, **Smt. Bharathi**, for granting me the opportunity to work in this significant project. The support has been a cornerstone in my journey of learning and development.

My sincere appreciation goes out to **Mr. Janardhan Reddy**, Scientist/Engg-SF, Spacecraft Checkout Group at **ISRO Satellite Integration and Test Establishment (ISITE)**, **U. R. RAO SATELLITE CENTRE**. His invaluable guidance, insightful suggestions, and unwavering encouragement have been pivotal in navigating the challenges of this internship. His mentorship not only guided me towards the successful completion of my internship but also instilled in me a deeper understanding and appreciation of the field.

Furthermore, I would like to express my profound gratitude to all the members of the **'SPACECRAFT CHECKOUT GROUP'**. Their knowledge, expertise, and willingness to share their experiences have greatly contributed to my learning. Their guidance has been a beacon of light, illuminating my path throughout this journey.

Lastly, I would like to acknowledge the role of every individual who directly or indirectly contributed to my personal and professional growth during this period. This internship has been a transformative experience, and the knowledge and skills acquired here will undoubtedly be pivotal in my future endeavors.

LIST OF ABBREVIATIONS

GUI	Graphical User Interface
RF	Radio Frequency
API	Application Programming Interface
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
SSD	Solid State Drive
RAM	Random Access Memory
CPU	Central Processing Unit
npm	Node Package Manager
CI/CD	Continuous Integration/Continuous Deployment
AG-Grid	A JavaScript Data Grid

ABSTRACT

The project titled "Database User Interface Design for Spacecraft Telemetry & Telecommand System RF Characterization Software" is dedicated to developing an intuitive and efficient user interface for managing spacecraft telemetry and telecommand operations, emphasizing RF (Radio Frequency) characterization. The system is specifically designed to automate the RF System characterization process.

The primary objective is to craft a user-friendly interface that streamlines complex data interactions. This enables users to effortlessly access, visualize, and manipulate RF system specifications and test data. The integration of data visualization tools, query builders, and real-time data monitoring features is a key aspect of this interface, customized to meet the unique demands of RF characterization crucial for effective spacecraft communication.

The project also integrates advanced database management techniques to efficiently manage large data volumes. This includes implementing optimized database schemas, indexing, and caching strategies for rapid data retrieval and processing.

Given the critical nature of spacecraft telemetry data, the software is developed with a strong emphasis on scalability and security. Features like user authentication, data encryption, and access control are incorporated to protect sensitive information.

Overall, this project significantly enhances the efficiency and effectiveness of spacecraft testing, especially in the RF characterization domain. It demonstrates the impactful application of user interface design principles in complex, specialized fields such as space technology, markedly contributing to the operational success of space missions.

INTRODUCTION

1.1 GENERAL INTRODUCTION

In the realm of space exploration and satellite communication, the role of efficient data management systems is paramount. The project, "Database User Interface Design for Spacecraft Telemetry & Telecommand System RF Characterization Software," addresses this critical need by developing a sophisticated user interface for managing and analyzing spacecraft telemetry and telecommand data, with a specific focus on RF (Radio Frequency) characterization.

Spacecraft telemetry involves the transmission of data from a spacecraft to Earth, encompassing vital information about the spacecraft's systems and environment. Telecommands, on the other hand, are instructions sent from Earth to the spacecraft to perform various operations. Both telemetry and telecommands are crucial for the successful operation of space missions, requiring precise and efficient handling.

The core challenge addressed by this project is the creation of a user interface that simplifies the complexity of handling RF System parameters. The user interface integrates tools for monitoring and analyzing RF signals, contributing to the effective validation of the spacecraft's communication system.

1.2 PROJECT OBJECTIVE

The primary objective of the "Database User Interface Design for Spacecraft Telemetry & Telecommand System RF Characterization Software" project is to develop a comprehensive and intuitive user interface that simplifies the management, analysis, and visualization of spacecraft telemetry and telecommand data, particularly focusing on RF (Radio Frequency) characterization.

This objective encompasses several key aspects:

1. **User Interface Development**: Designing an easy-to-navigate, efficient, and user-friendly interface that enables users to interact seamlessly with complex RF system parameters.
2. **Data Management Efficiency**: Implementing advanced data management strategies to handle large volumes of data efficiently, ensuring quick retrieval and processing.
3. **RF Characterization Integration**: Incorporating specialized tools and functionalities for the analysis and monitoring of RF signals, which are crucial for effective spacecraft communication.
4. **Scalability and Security**: Ensuring that the system is scalable to accommodate future advancements in spacecraft technology and secure to protect sensitive mission data.

SYSTEM PROPOSAL

2.1 EXISTING SYSTEM

The current systems in place for in the context of RF characterization, typically involve complex software platforms that are designed to handle a vast amount of technical data. These systems are often developed with a focus on functionality and data accuracy, catering to the technical requirements of spacecraft operations. However, they might not prioritize user interface design and user experience.

2.1.1 DISADVANTAGES

1. Lack of User-Friendliness: One of the primary drawbacks is that these systems are often not user-friendly. They may have steep learning curves due to complex interfaces, making it difficult for users, especially those who are not deeply familiar with the system, to navigate and utilize the software efficiently.

2. Integration of Operations: Current systems might operate in silos, lacking a unified interface that integrates various aspects of spacecraft operations. This can lead to a disjointed experience and can complicate the process.

3. Limited Customization and Flexibility: The existing systems may offer limited options for customization and flexibility, hindering the ability of users to tailor the interface and functionalities according to their specific needs.

4. Inadequate Visualization Tools: Effective data visualization is key for interpreting test data. Existing systems might lack advanced visualization tools, making it challenging to analyze and interpret data effectively.

The project "Database User Interface Design for Spacecraft Telemetry & Tele Command System RF Characterization Software" aims to overcome these disadvantages by introducing a more user-friendly, efficient, and integrated system. The core challenge is the creation of a user interface that simplifies handling RF Systems data.

2.2 PROPOSED SYSTEM

The "Database User Interface Design for Spacecraft Telemetry & Telecommand System RF Characterization Software" project proposes a new system that significantly enhances the user experience and operational efficiency during ground testing of TTC-RF systems. This system is designed with a focus on user-friendliness, efficient data management and robust security measures.

2.2.1 ADVANTAGES

1. Improved User Experience: The intuitive interface significantly reduces the learning curve, making the system more accessible to a broader range of users.

2. Efficient Operation: Enhanced data management leads to quicker data processing and retrieval, improving overall operational efficiency.

3. Streamlined Data Handling: The unified interface simplifies the handling of diverse operations, reducing the likelihood of errors and improving decision-making processes.

4. Future-Ready Design: Scalability ensures that the system remains relevant and effective as mission demands evolve.

5. Data Security and Integrity: With robust security measures, the system ensures the safety and confidentiality of mission-critical data.

6. Personalized User Experience: Customization options allow users to tailor the system to their specific requirements, enhancing usability.

7. Effective Data Analysis: Advanced visualization tools enable users to derive meaningful insights from complex data sets more efficiently.

Overall, the proposed system aims to revolutionize the way spacecraft telemetry and telecommand operations are managed, making them more user-friendly, efficient, and secure, thereby contributing significantly to the success of space missions.

2.3 LITERATURE SURVEY

A literature survey of these technologies helps in understanding their applications, advantages, and how they can be effectively used in the project.

1. Vue.js:

- Overview: Vue.js is a progressive JavaScript framework used for building user interfaces. It is designed to be incrementally adoptable, making it an ideal choice for single-page applications and more complex interfaces.
- Relevant Literature: Studies and articles often focus on Vue.js's reactivity system, component-based architecture, and its use case in developing dynamic web interfaces. The framework's ease of integration with other libraries and projects makes it a popular choice for modern web development.

2. Python:

- Overview: Python is a high-level, interpreted programming language known for its readability, simplicity, and vast library support. It is widely used in various domains, including web development, data analysis, artificial intelligence, and scientific computing.
- Relevant Literature: Research papers and technical articles on Python often discuss its application in data processing, scripting, and automation. Its role in backend development, particularly in handling database operations and server-side logic, is also well-documented.

3. MongoDB:

- Overview: MongoDB is a NoSQL database that provides high performance, high availability, and easy scalability. It uses a document-oriented data model, which makes it suitable for storing large volumes of unstructured data.
- Relevant Literature: Literature on MongoDB typically explores its flexible schema design, scalability features, and performance metrics. Studies often compare it with traditional relational databases, highlighting its advantages in handling big data and real-time analytics.

For the project, these technologies offer the following advantages:

- **Vue.js** allows for creating a responsive and dynamic user interface, enhancing user experience and interaction with the system.
- **Python**, with its simplicity and powerful libraries, is ideal for backend development, data manipulation, and integration with other systems like MongoDB.
- **MongoDB** offers a scalable and flexible database solution, capable of handling the large and varied datasets typical in spacecraft telemetry and telecommand operations.

Combining these technologies, the project can leverage Vue.js for frontend development, Python for backend operations and data processing, and MongoDB for efficient data storage and retrieval. This combination promises a robust, scalable, and user-friendly system for managing complex spacecraft operations data.

SYSTEM DETAILS

3.1 ARCHITECTURE DETAILS

- **Frontend Layer**: Display Vue.js as the frontend layer, showcasing elements like the User Interface, Data Visualization Components, and Interactive Dashboard.
- **Backend Layer**: Illustrate Python as the backend layer. Include modules for Data Processing, API Endpoints, and Integration Logic.
- **Database Layer**: Represent MongoDB, highlighting its role in Data Storage, which includes Telemetry Data, Telecommand Data, and RF Characterization Data.

3.2. FLOW DETAILS

- **User Interaction**: Start with User Interaction in the Vue.js interface.
- **Data Request/Command**: Follow the flow to Data Request (Telemetry Data View/Edit) or Command Issuance (Telecommands).
- **Processing Request**: Show the flow going to Python backend where the data is processed or commands are interpreted.
- **Database Interaction**: From the backend, the flow moves to MongoDB for data retrieval or update.
- **Response Generation**: Back to the Python backend for processing the response.
- **Data Visualization/Command Confirmation**: Flow returns to the Vue.js frontend for displaying data or confirming command execution.

IMPLEMENTATION

4.1 MODULES AND DESCRIPTION

For the project "Database User Interface Design for Spacecraft Telemetry & Tele Command System RF Characterization Software," the system can be broken down into several key modules. Each module serves a specific function and works in coordination with others to achieve the overall objective of the system. Here is an overview of the main modules and their descriptions:

1. User Interface Module (Frontend)

- Description: This module, developed using Vue.js, is responsible for the user-facing part of the application. It includes the design and implementation of the graphical user interface (GUI), ensuring it is intuitive, responsive, and user-friendly. This module handles user interactions, data visualization, and presentation of telemetry and telecommand data.
- Key Features: Interactive dashboards, real-time data display, customizable views, and intuitive navigation.

2. Data Processing Module (Backend)

- Description: Developed in Python, this module is the core of the application's backend. It processes incoming telemetry data, executes telecommand operations, and handles the logic for RF characterization. It also manages data transformations, computations, and serves as a bridge between the frontend and the database.
- Key Features: Data validation, transformation, execution of logic, and API endpoint creation.

3. Database Management Module

- Description: This module utilizes MongoDB for data storage and management. It is responsible for storing, retrieving, and managing telemetry and telecommand data efficiently. It handles data indexing, query optimization, and ensures data integrity and security.

- Key Features: Efficient data storage, high-performance querying, data integrity, and backup mechanisms.

4. Security and Authentication Module

- Description: This module is crucial for ensuring the security and integrity of the system. It manages user authentication, authorization, data encryption, and protects against unauthorized access.
- Key Features: User login/logout, role-based access control, data encryption, and security audits.

5. Integration and Communication Module

- Description: This module facilitates communication and integration with external systems, such as spacecraft telemetry systems, other databases, or third-party services. It ensures seamless data exchange and system interoperability.
- Key Features: API integration, data exchange protocols, external system interfacing.

6. Reporting and Analytics Module

- Description: Responsible for generating comprehensive reports and analytics from the processed data. This module provides insights into the spacecraft's performance and helps in decision-making.
- Key Features: Custom reports, performance metrics, analytics.

Each module is designed to function both independently and as part of the larger system, ensuring modularity, ease of maintenance, and future scalability.

4.2 MY WORK

IMPLEMENTATION OVERVIEW AND DESCRIPTION

User Authentication & Access Control: The implementation commenced with a robust user authentication mechanism, seamlessly blending Vue.js and FastAPI. This strategic combination not only ensured a secure login process but also laid the foundation for a flexible role-based access control system. Through this system, user permissions were precisely defined based on designated roles such as administrators and standard users. The intricate interplay of Vue.js and FastAPI allowed for a smooth and secure user authentication and access control framework.

User Management Features: Within the user management domain, a user-friendly interface was crafted, specifically tailored for administrators. This interface facilitated the addition and management of users with an emphasis on simplicity and efficiency. Security was a top priority, leading to the implementation of robust methods for secure password changes and profile management. The goal was to empower administrators with tools that not only simplify their tasks but also ensure the security of user data and account details.

Test Profile for Receivers and Transformers: The development efforts extended to the efficient handling of test profiles, a critical aspect of spacecraft testing. Functionality was engineered to empower administrators in managing test profiles seamlessly. This included features for the streamlined management of test profile names linked to both receivers and transponders. The implementation aimed to enhance the overall testing process, providing administrators with tools to efficiently organize and oversee test profiles.

Functionality Integration: A user-centric approach was adopted for feedback mechanisms within the application. Real-time feedback became an integral part of the user experience. Toast messages were strategically integrated to provide instant feedback during user interactions, ensuring that users enter correct information.

Additionally, inline messages were strategically placed throughout the application, fostering effective communication within the system. This emphasis on user feedback aimed to enhance user engagement and satisfaction.

This comprehensive approach to implementation ensured that each component played a pivotal role in achieving the project's overarching goals of security, user-friendliness, and efficient spacecraft testing operations.

CODING APPROACH AND METHEDOLOGIES

Password Security with Cryptographic Hashing:

In the realm of user security, a meticulous approach was adopted to fortify sensitive information, particularly in password management. The system incorporated a robust hashing mechanism, ensuring that passwords were never stored in their raw form. Instead, they were transformed into irreversible hash values. This cryptographic strategy, implemented using Python within the FastAPI backend, provided an additional layer of protection against potential security breaches. The integration of secure hashing algorithms aligned with industry best practices, enhancing the overall security infrastructure.

Role-Based User Management:

User management operations, encompassing tasks such as user addition and deletion, were rigorously structured to adhere to the principle of least privilege. The system ensured that only administrators possessed the authority to execute such critical operations, preventing unauthorized modifications. This administrative control was enforced through role-based access control, allowing administrators exclusive rights to user management functionalities. The backend, empowered by FastAPI, orchestrated these operations with precision, ensuring compliance with stringent security protocols.

Dynamic Frontend Updates:

The collaborative synergy between the backend and frontend was evident in the seamless integration of user management updates within the Vue.js frontend. When administrators performed actions, such as adding or deleting users, these changes were dynamically reflected in the frontend interface. Vue.js, renowned for its reactive and component-based architecture, facilitated real-time updates without necessitating manual interventions. This dynamic interaction between the backend and frontend contributed to an intuitive user experience, elevating the overall usability and functionality of the system.

In essence, the password management system implemented cryptographic hashing for heightened security, user management operations were confined to administrators through role-based access control, and the harmonious integration between the backend and Vue.js frontend ensured that updates seamlessly propagated for a cohesive and responsive user interface.

SYSTEM REQUIREMENTS

For the project "Database User Interface Design for Spacecraft Telemetry & Telecommand System RF Characterization Software," there are specific hardware and software requirements to ensure optimal performance and functionality. These requirements are essential for developing, deploying, and running the system efficiently.

5.1 HARDWARE REQUIREMENTS

1. Server Specifications:

- Processor: Multi-core processor (e.g., Intel Xeon, AMD Ryzen) for efficient handling of concurrent processes.
- Memory: Minimum 16GB RAM, preferable 32GB or more for handling large datasets and complex operations.
- Storage: High-speed SSD with at least 1TB capacity for fast data access and storage. Additional backup storage solutions.
- Network Interface: Gigabit Ethernet for fast data transfer and communication.

2. Client (User Interface) Specifications:

- Processor: Modern CPU capable of running modern web browsers efficiently (e.g., Intel Core i5/i7, AMD Ryzen 5/7).
- Memory: Minimum 8GB RAM.
- Display: High-resolution display (1080p or higher) for clear data visualization.
- Network Connectivity: Reliable internet connection for accessing the server and cloud resources.

3. Development and Testing Equipment:

- Computers: Adequate computers for developers and testers with specifications suitable for software development and testing tasks.

5.2 SOFTWARE REQUIREMENTS

1. Frontend Development:

- Framework: Vue.js.
- Development Tools: Node.js, npm (Node Package Manager), and supporting libraries for Vue.js.

2. Backend Development:

- Programming Language: Python.
- Frameworks/Libraries: FastAPI for API development, Pandas for data manipulation, and other Python libraries as needed.

3. Database:

- Database System: MongoDB.
- Tools: MongoDB management tools like MongoDB Compass or Robo 3T for database administration.

4. Operating System:

- Server: A stable Linux distribution like Ubuntu Server, CentOS, or Red Hat Enterprise Linux.
- Client and Development Machines: Compatibility with various operating systems including Windows, macOS, and Linux distributions.

5. Security Software:

- Firewall: Configured at both server and client levels.
- Antivirus/Anti-malware: Standard antivirus software for all machines.
- Data Encryption Tools: For securing data in transit and at rest.

6. Testing and Analysis Tools:

- Unit Testing Frameworks: PyTest for Python, Jest for Vue.js.
- Performance Testing Tools: JMeter, LoadRunner, or similar.

5.3 SOFTWARE DESCRIPTION

The "Database User Interface Design for Spacecraft Telemetry & Tele Command System RF Characterization Software" project leverages a range of software technologies, each playing a critical role in the system's functionality and performance. Here's a description of the key software components used in this project:

1. Vue.js

- Type: JavaScript Framework

- Description: Vue.js is a progressive framework used for building user interfaces. Its core focus is on the view layer, making it easy to integrate with other libraries or existing projects. Vue.js is known for its simplicity, versatility, and fine-grained reactivity system that makes it straightforward to develop interactive and dynamic web applications.

2. Python

- Type: Programming Language

- Description: Python is a high-level, interpreted programming language renowned for its clear syntax and readability. It supports multiple programming paradigms and comes with a comprehensive standard library. In this project, Python is primarily used for backend development, including server-side logic, data manipulation, and interaction with the database.

3. MongoDB

- Type: NoSQL Database

- Description: MongoDB is a document-oriented NoSQL database used for high volume data storage. It is particularly known for its scalability and flexibility in dealing with diverse data types. MongoDB stores data in JSON-like documents, which makes it a good fit for applications that require real-time analytics and high-speed logging and updating of data.

4. Node.js and npm

- Type: JavaScript Runtime and Package Manager
- Description: Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine, enabling JavaScript to be used for server-side scripting. npm (Node Package Manager) is used for managing JavaScript packages. In this project, Node.js and npm are utilized for managing dependencies and running build scripts for the Vue.js frontend.

5. Git

- Type: Version Control System
- Description: Git is a distributed version control system used for tracking changes in source code during software development. It supports distributed workflows, enhances collaboration, and provides a historical record of the project development process.

6. Linux (Ubuntu Server, CentOS, etc.)

- Type: Operating System
- Description: A Linux distribution is used as the server operating system due to its stability, security, and performance. Distributions like Ubuntu Server or CentOS offer robustness and are widely used in server environments.

These software components are integral to building a responsive, efficient, and scalable system for managing spacecraft telemetry and telecommand operations. Each component has been selected for its proven reliability and compatibility in professional software development environments.

CONCLUSION

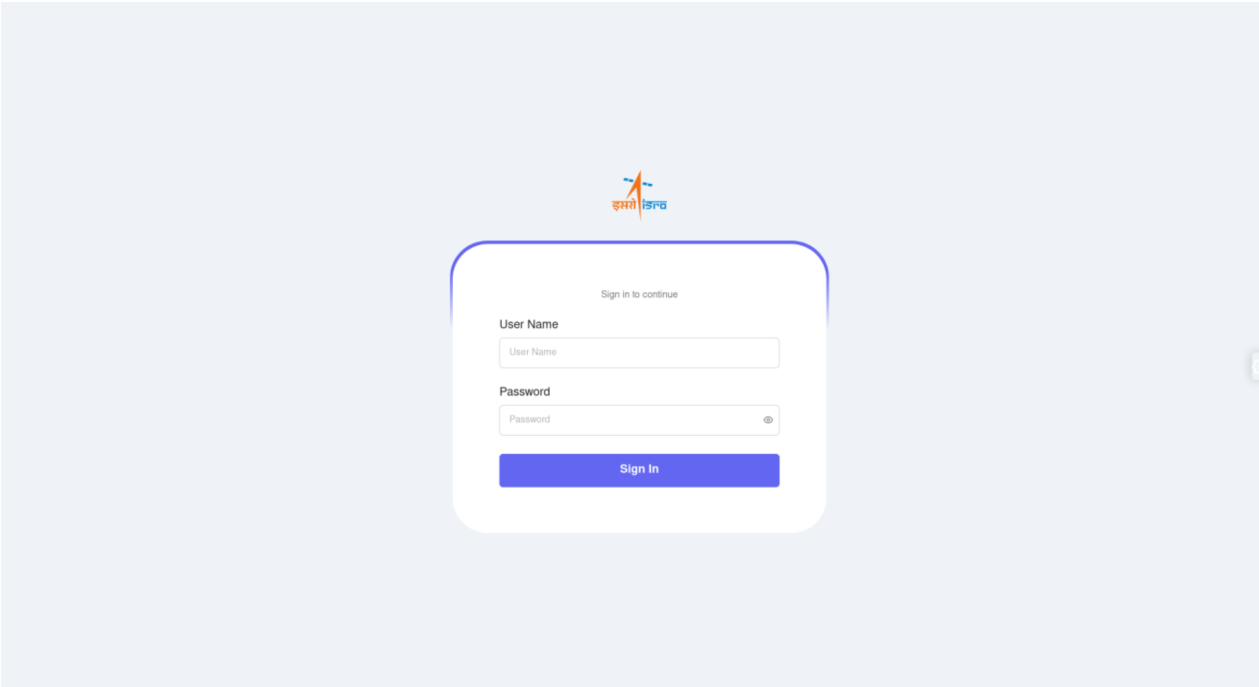
In conclusion, the "Database User Interface Design for Spacecraft Telemetry & Tele Command System RF Characterization Software" project represents a significant advancement in the field of spacecraft testing. By integrating cutting-edge technologies such as Vue.js for the frontend interface, Python for backend processing, and MongoDB for database management, the project offers a comprehensive and robust solution tailored to the specific needs of spacecraft telemetry and telecommand operations.

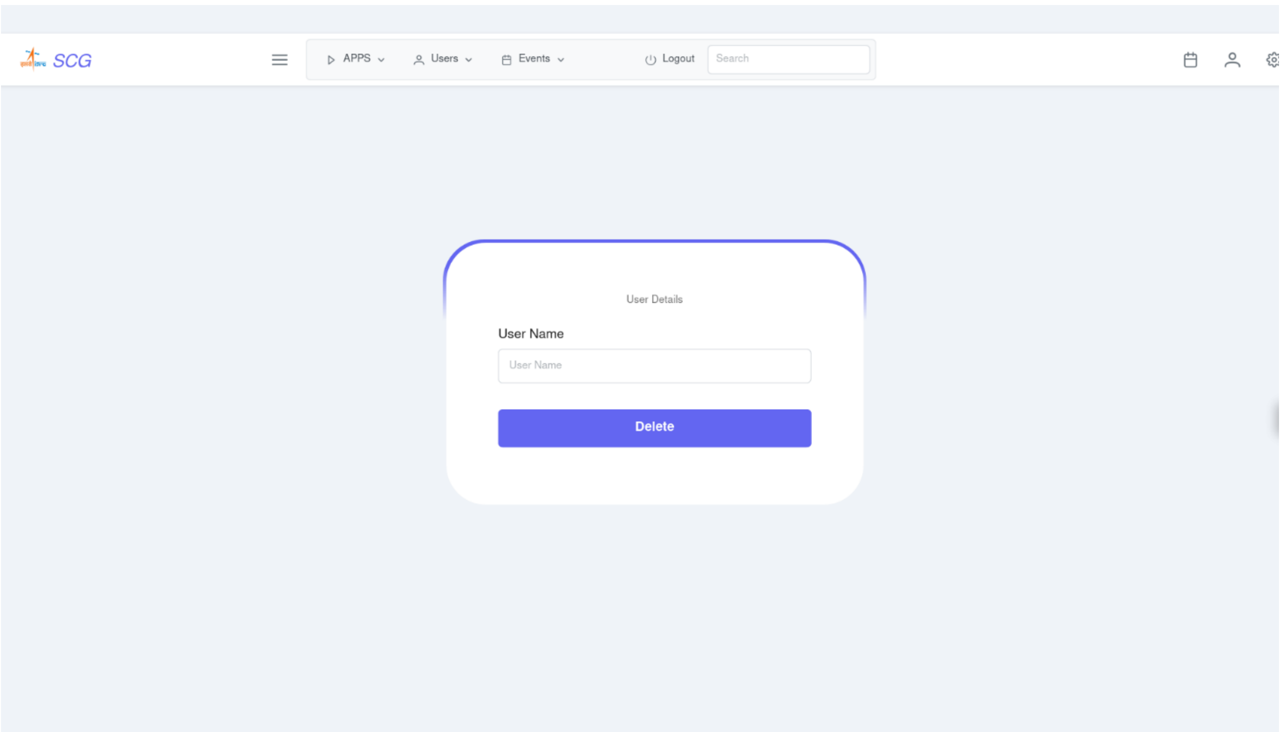
The project addresses several key challenges in the current systems, including the need for a user-friendly interface, efficient data management, and enhanced security measures. The proposed system stands out with its intuitive user interface, making complex data interactions more accessible to a wider range of users, regardless of their technical expertise. The integration of advanced data management and visualization tools ensures that vast amounts of telemetry data can be processed and analyzed efficiently and effectively.

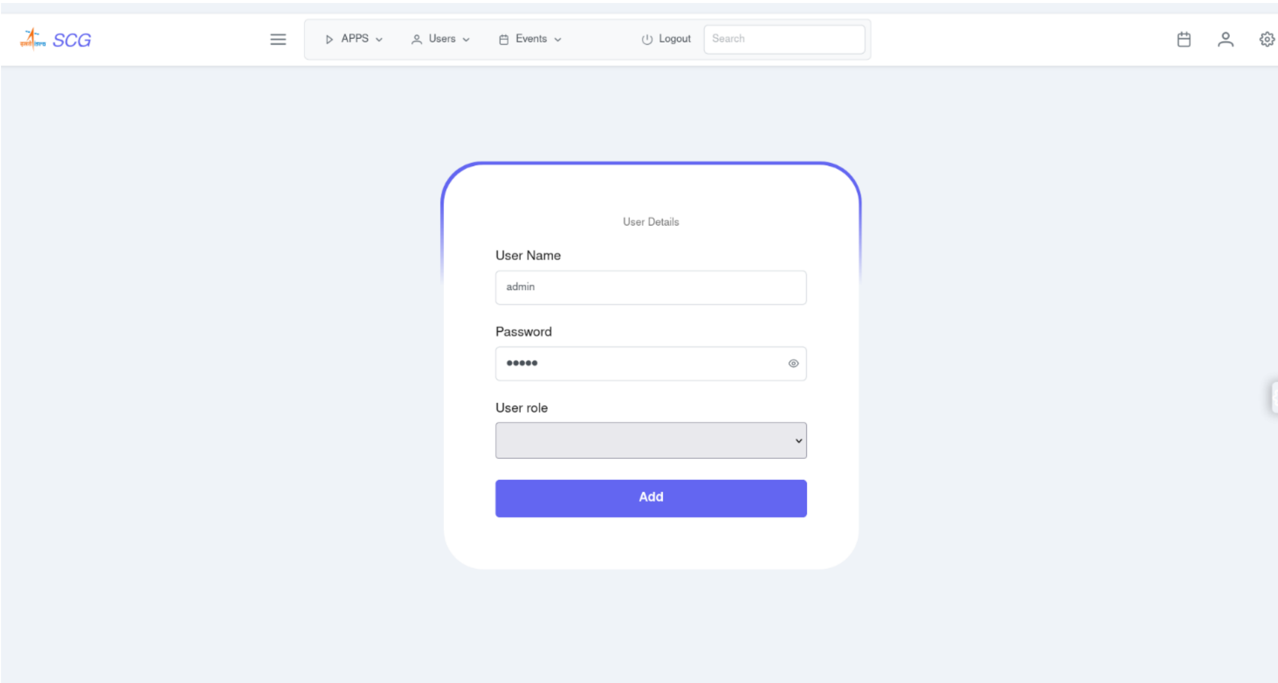
Moreover, the focus on RF characterization provides critical insights into the spacecraft's communication systems, essential for the success of any space mission. The system's scalable architecture ensures it can adapt to evolving mission requirements, and the stringent security protocols safeguard sensitive mission data.

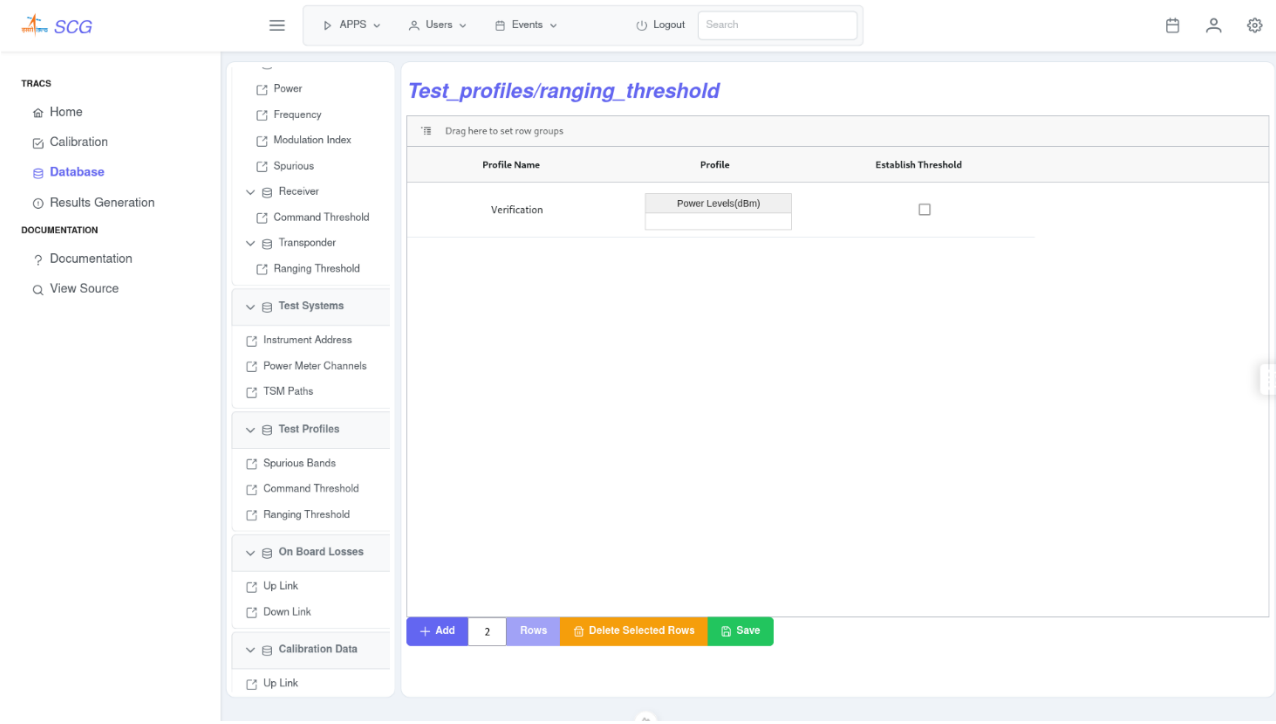
In summary, this project not only elevates the standard of spacecraft telemetry and telecommand systems but also sets a precedent for future developments in this domain. The implementation of this system is expected to lead to more efficient, secure, and successful space missions, underscoring the pivotal role of advanced software solutions in the ever-evolving realm of space exploration and communication.

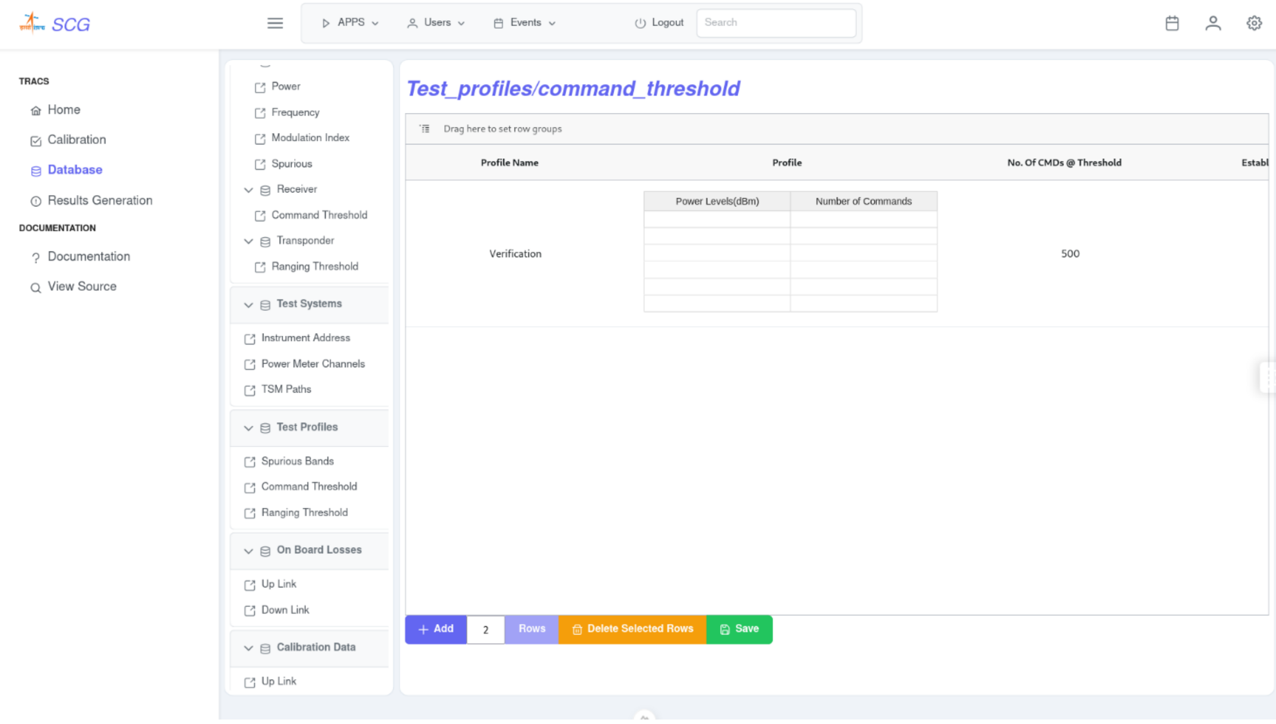
SAMPLE SCREENSHOT

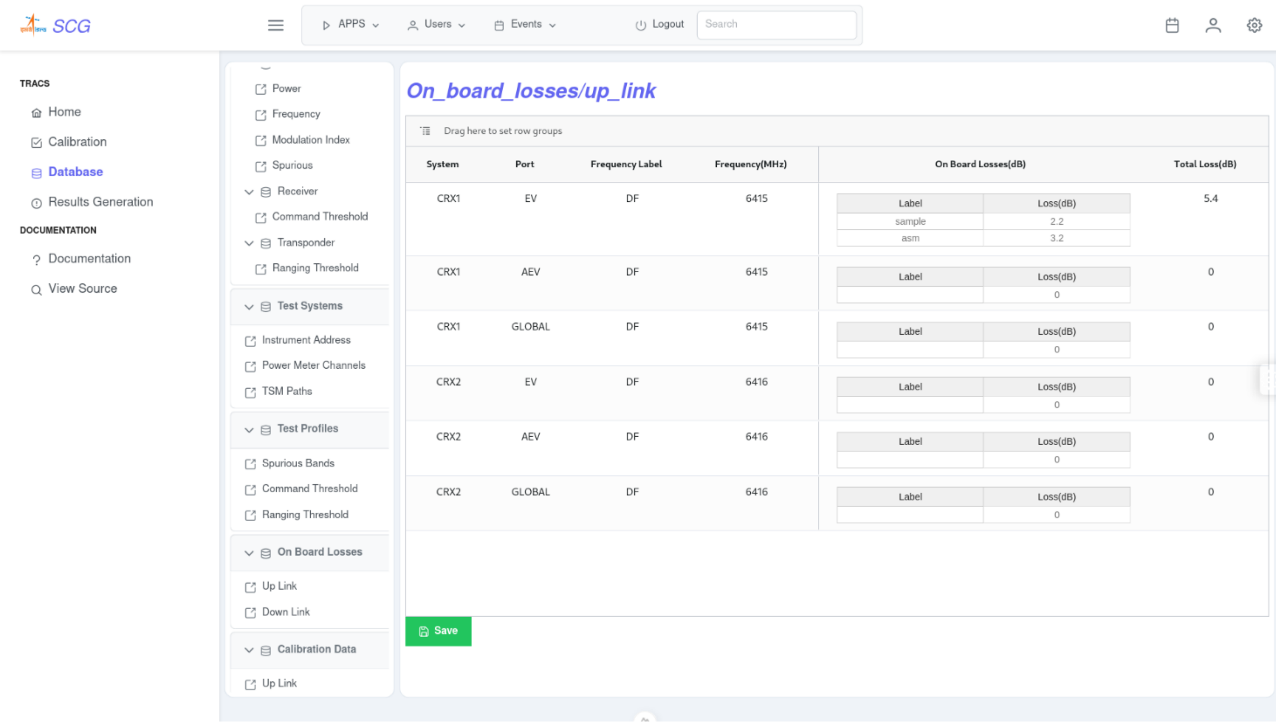


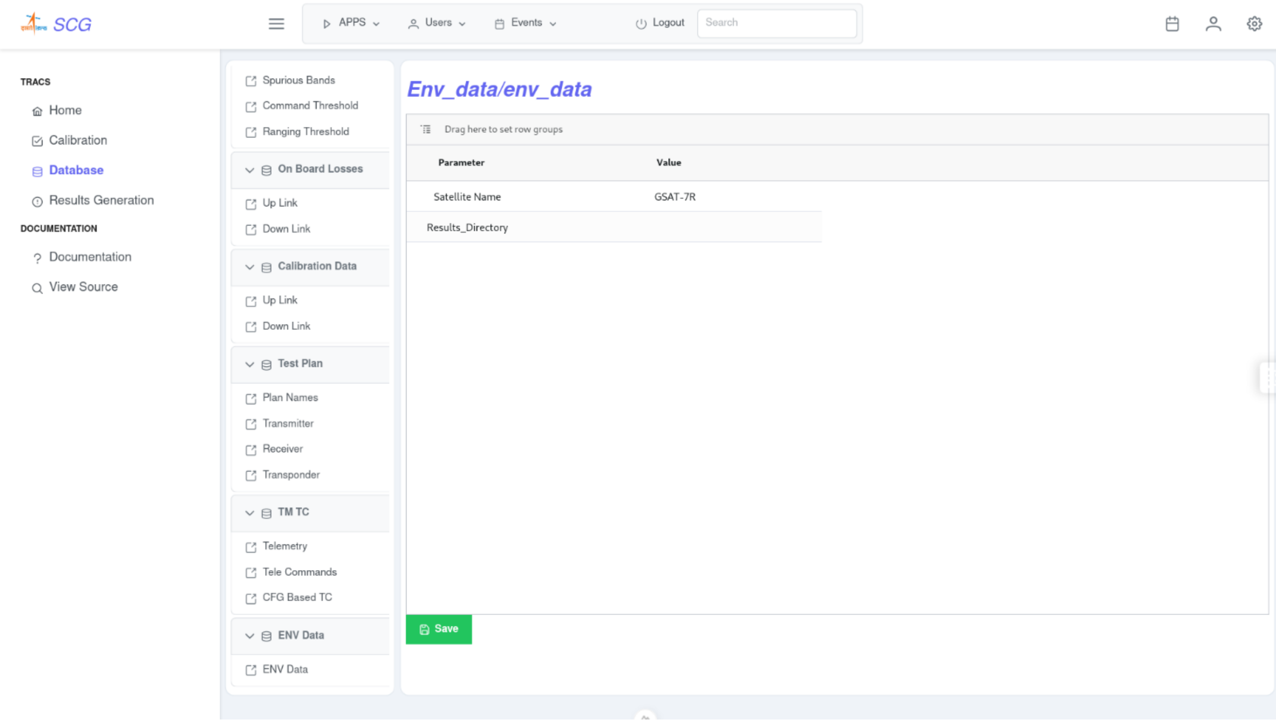


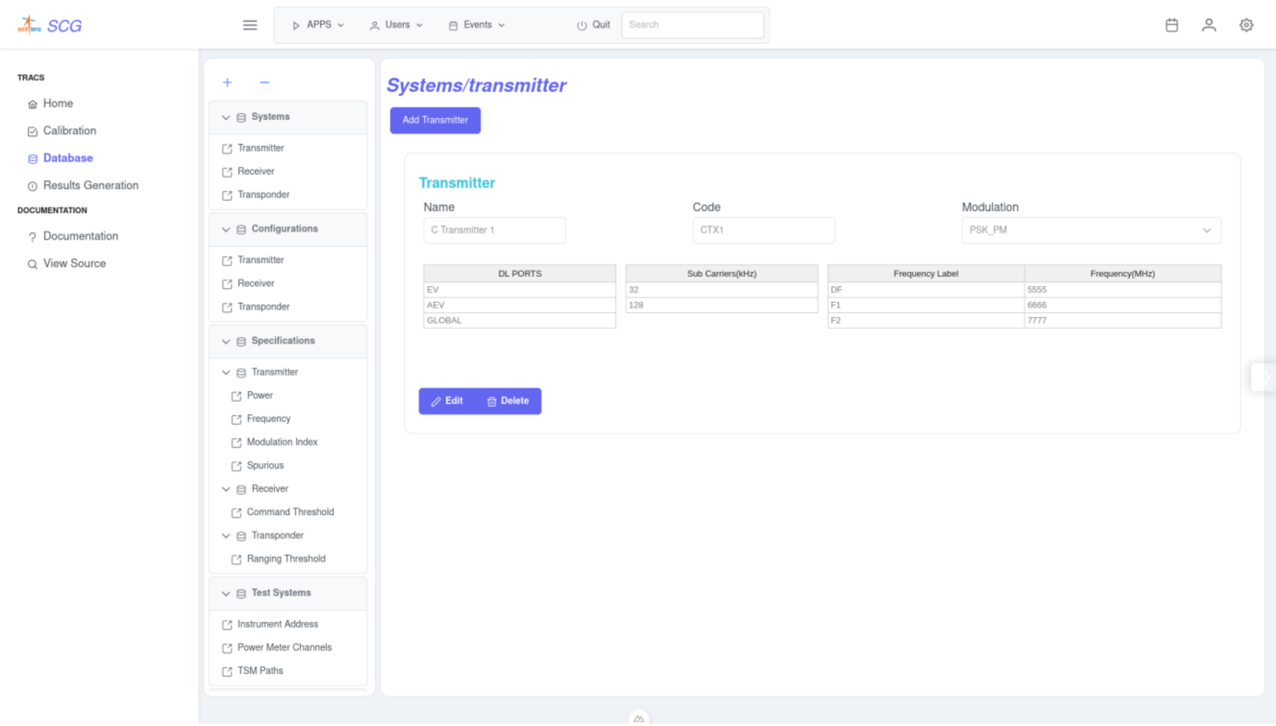

SCG
APPS
Users
Events
Logout
Search
User Details
User Name
admin
Password
User role
Add

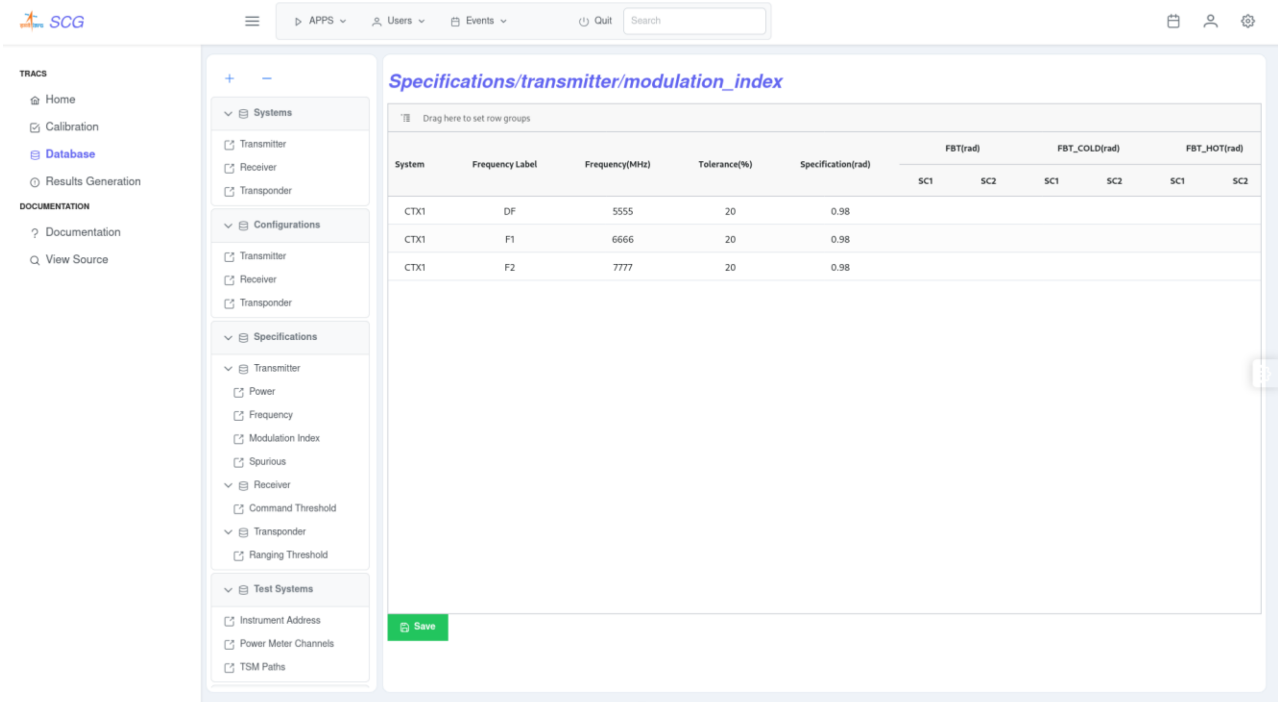

SCG
APPS
Users
Events
Logout
Search
TRACS
Home
Calibration
Database
Results Generation
DOCUMENTATION
Documentation
View Source
Power
Frequency
Modulation Index
Spurious
Receiver
Command Threshold
Transponder
Ranging Threshold
Test Systems
Instrument Address
Power Meter Channels
TSM Paths
Test Profiles
Spurious Bands
Command Threshold
Ranging Threshold
On Board Losses
Up Link
Down Link
Calibration Data
Up Link
Test_profiles/ranging_threshold
Drag here to set row groups
Profile Name
Profile
Establish Threshold
Verification
Power Levels(dBm)
Add
2
Rows
Delete Selected Rows
Save


SCG
APPS
Users
Events
Logout
Search
TRACS
Home
Calibration
Database
Results Generation
DOCUMENTATION
Documentation
View Source
Power
Frequency
Modulation Index
Spurious
Receiver
Command Threshold
Transponder
Ranging Threshold
Test Systems
Instrument Address
Power Meter Channels
TSM Paths
Test Profiles
Spurious Bands
Command Threshold
Ranging Threshold
On Board Losses
Up Link
Down Link
Calibration Data
Up Link
Test_profiles/command_threshold
Drag here to set row groups
Profile Name
Profile
No. Of CMDs @ Threshold
Power Levels(dBm)
Number of Commands
Verification
500
Add
2
Rows
Delete Selected Rows
Save


SCG
APPS
Users
Events
Logout
Search
TRACS
Home
Calibration
Database
Results Generation
DOCUMENTATION
Documentation
View Source
Power
Frequency
Modulation Index
Spurious
Receiver
Command Threshold
Transponder
Ranging Threshold
Test Systems
Instrument Address
Power Meter Channels
TSM Paths
Test Profiles
Spurious Bands
Command Threshold
Ranging Threshold
On Board Losses
Up Link
Down Link
Calibration Data
Up Link
On_board_losses/up_link
Drag here to set row groups
System
Port
Frequency Label
Frequency(MHz)
On Board Losses(dB)
Total Loss(dB)
CRX1 EV DF 6415 5.4
Label Loss(dB)
sample 2.2
asm 3.2
CRX1 AEV DF 6415 0
CRX1 GLOBAL DF 6415 0
CRX2 EV DF 6416 0
CRX2 AEV DF 6416 0
CRX2 GLOBAL DF 6416 0
Save

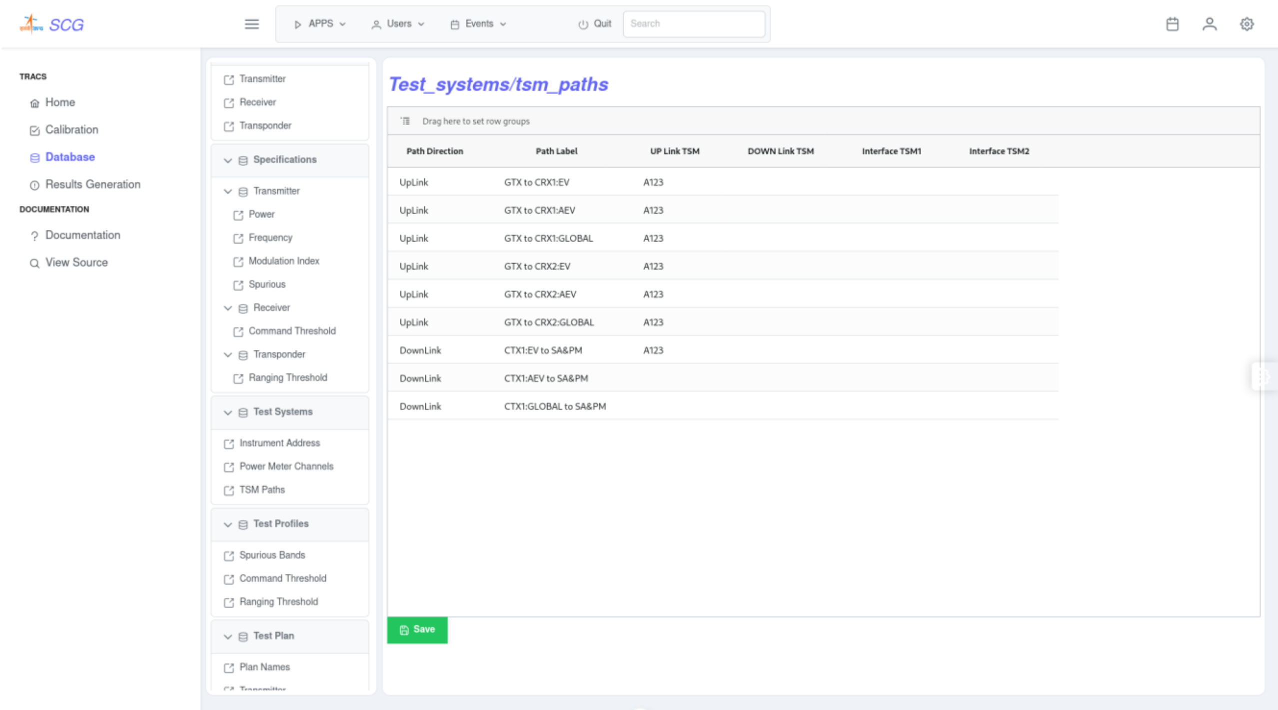

SCG
APPS
Users
Events
Logout
Search
TRACS
Home
Calibration
Database
Results Generation
DOCUMENTATION
Documentation
View Source
Spurious Bands
Command Threshold
Ranging Threshold
On Board Losses
Up Link
Down Link
Calibration Data
Up Link
Down Link
Test Plan
Plan Names
Transmitter
Receiver
Transponder
TM TC
Telemetry
Tele Commands
CFG Based TC
ENV Data
ENV Data
Env_data/env_data
Drag here to set row groups
Parameter
Value
Satellite Name
GSAT-7R
Results_Directory
Save


SCG
APPS
Users
Events
Quit
Search
TRACS
Home
Calibration
Database
Results Generation
DOCUMENTATION
Documentation
View Source
Systems
Transmitter
Receiver
Transponder
Configurations
Transmitter
Receiver
Transponder
Specifications
Transmitter
Power
Frequency
Modulation Index
Spurious
Receiver
Command Threshold
Transponder
Ranging Threshold
Test Systems
Instrument Address
Power Meter Channels
TSM Paths
Systems/transmitter
Add Transmitter
Transmitter
Name
Code
Modulation
C Transmitter 1
CTX1
PSK_PM
DL PORTS
Sub Carriers(kHz)
Frequency Label
Frequency(MHz)
EV
AEV
GLOBAL
32
128
DF
F1
F2
5555
6666
7777
Edit
Delete


SCG
APPS
Users
Events
Quit
Search
TRACS
Home
Calibration
Database
Results Generation
DOCUMENTATION
Documentation
View Source
Systems
Transmitter
Receiver
Transponder
Configurations
Transmitter
Receiver
Transponder
Specifications
Transmitter
Power
Frequency
Modulation Index
Spurious
Receiver
Command Threshold
Transponder
Ranging Threshold
Test Systems
Instrument Address
Power Meter Channels
TSM Paths
Specifications/transmitter/modulation_index
Drag here to set row groups

System	Frequency Label	Frequency(MHz)	Tolerance(%)	Specification(rad)	FBT(rad)		FBT_COLD(rad)		FBT_HOT(rad)	
					SC1	SC2	SC1	SC2	SC1	SC2
CTX1	DF	5555	20	0.98						
CTX1	F1	6666	20	0.98						
CTX1	F2	7777	20	0.98						

Save


SCG
APPS
Users
Events
Quit
Search
TRACS
Home
Calibration
Database
Results Generation
DOCUMENTATION
Documentation
View Source
Transmitter
Receiver
Transponder
Specifications
Transmitter
Power
Frequency
Modulation Index
Spurious
Receiver
Command Threshold
Transponder
Ranging Threshold
Test Systems
Instrument Address
Power Meter Channels
TSM Paths
Test Profiles
Spurious Bands
Command Threshold
Ranging Threshold
Test Plan
Plan Names
Test_systems/tsm_paths
Drag here to set row groups

Path Direction	Path Label	UP Link TSM	DOWN Link TSM	Interface TSM1	Interface TSM2
UpLink	GTX to CRX1:EV	A123			
UpLink	GTX to CRX1:AEV	A123			
UpLink	GTX to CRX1:GLOBAL	A123			
UpLink	GTX to CRX2:EV	A123			
UpLink	GTX to CRX2:AEV	A123			
UpLink	GTX to CRX2:GLOBAL	A123			
DownLink	CTX1:EV to SA&PM	A123			
DownLink	CTX1:AEV to SA&PM				
DownLink	CTX1:GLOBAL to SA&PM				

Save

REFERENCES

1. **Vue.js Documentation**: Vue.js Developers. *Vue.js Guide*.

   Link: https://vuejs.org/guide/introduction.html

2. **Python Language Reference**: Van Rossum, G., & Drake, F. L. (2009). *Python Reference Manual*.

   Link: Python official documentation, https://docs.python.org/3/

3. **MongoDB Usage and Applications**: MongoDB, Inc. *MongoDB Documentation*. Link: https://www.mongodb.com/docs/

4. **Data Management in Spacecraft Telemetry**: Sk Ibrahim, A Ahamed , MAE Zeidan (2018). *Machine learning methods for* ***spacecraft telemetry*** *mining*

5. **RF Characterization Techniques**: Zhengxiang Ma; A.J. Becker; P. Polakos; H. Huggins; J. Pastalan; Hui Wu; K. Watts; Y.H. Wong; P. Mankiewich RF measurement technique for characterizing thin dielectric films

6. **User Interface Design Principles**: Nielsen, J., & Norman, D. (2010). Gestural interfaces: a step backward in usability.

ADDITIONAL REFERENCES

AG-Grid Documentation: AG-Grid Team. *AG-Grid Documentation*. AG-Grid official website

AG-Grid is a widely-used JavaScript grid library that provides a powerful solution for displaying and manipulating large sets of data in web applications.

PrimeVue UI Components: PrimeTek. *PrimeVue Guide*. PrimeVue official website

PrimeVue offers a rich set of UI components for Vue.js applications.