GROUP - 21

HOUSE PRICE PREDICTION

Phase 4: Develop a User Interface

Team Members:

Gurramkonda Naga Prasanna - 11706408

Supriya Ravilla - 11644767

Tharun Kuravadi Sathish Babu - 11659825

Vamsi Krishna Kanchukommala - 11666316

Yeshwanth Sai Kumar Thotakura - 11648659

Architectural Overview and Communication Flow:

The Mumbai House Price Prediction System implements a layered architecture with clear separation of concerns. The communication diagrams reveal three distinct layers: Front-End, Back-End, and Data Layer. The Front-End Layer consists of an intuitive web interface where users input property details such as location (selected from Mumbai areas), area in square feet, number of bedrooms, floor number, property type, and various amenities like gymnasium, car parking, indoor games, and jogging track. The UI components are designed to collect accurate data while providing a seamless user experience. The form validation ensures data integrity before transmission to the backend.

The Back-End Layer demonstrates sophisticated processing capabilities through multiple interconnected components. The Flask API server serves as the gateway, handling incoming requests and managing responses. The Model Manager acts as the orchestrator, coordinating between different components and managing the prediction workflow. Two machine learning models - Random Forest and Decision Tree - are implemented to provide reliable price predictions. The Data Preprocessor handles crucial tasks such as feature scaling, categorical encoding (especially for location and property type), and data transformation. One notable component is the Feature Importance Calculator, which provides transparency by showing

users how different features influence the predicted price, enhancing trust in the system's predictions.

Primary Use Cases and Implementation:

1. Property Price Prediction:
    - User inputs property details through the web interface
    - System validates and processes input data
    - Both Random Forest and Decision Tree models generate predictions
    - System displays predicted price along with a confidence range
    - Feature importance scores show impact of each property characteristic
2. Property Comparison:
    - Users can save multiple property predictions
    - System enables side-by-side comparison of different properties
    - Comparison includes price predictions and feature differences
    - Helps users make informed decisions about property selection
3. Market Analysis:
    - System provides feature importance analysis
    - Users can understand which factors most affect property prices
    - Location and area typically show high importance scores
    - Helps users optimize property selection based on key features

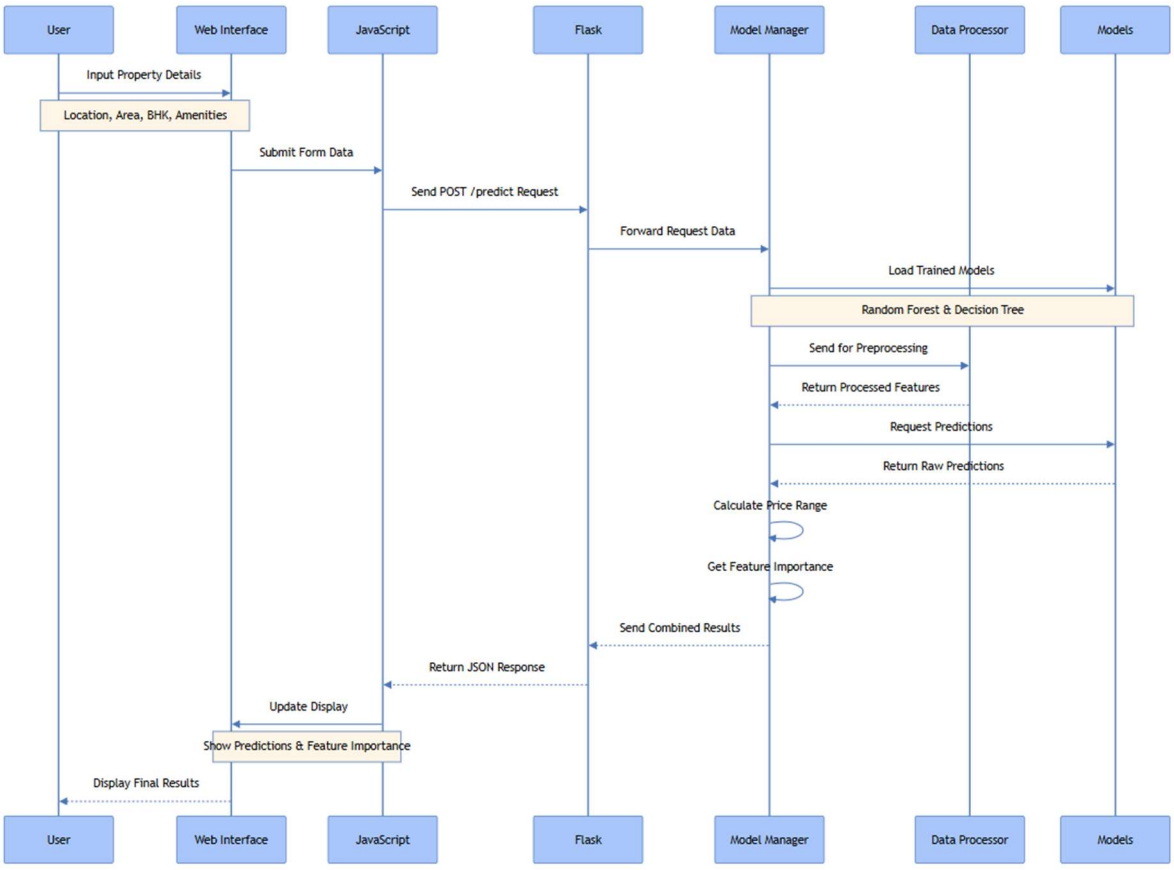
Data Flow and Processing:

The communication flow begins when a user submits property details (step 1) through the web interface. The formatted request is sent to the Flask server (step 2-3), which forwards it to the Model Manager. The Data Preprocessor (step 4-5) transforms the input data, handling categorical variables and scaling numerical features. Both machine learning models process the prepared data (step 6-8), accessing saved model parameters from the Data Layer. The Model Manager aggregates predictions and calculates feature importance (step 9). Finally, the

processed results flow back through the API to the user interface (step 10-11), displaying the predicted price, model-specific predictions, and feature importance analysis.

Technical Implementation and Error Handling:

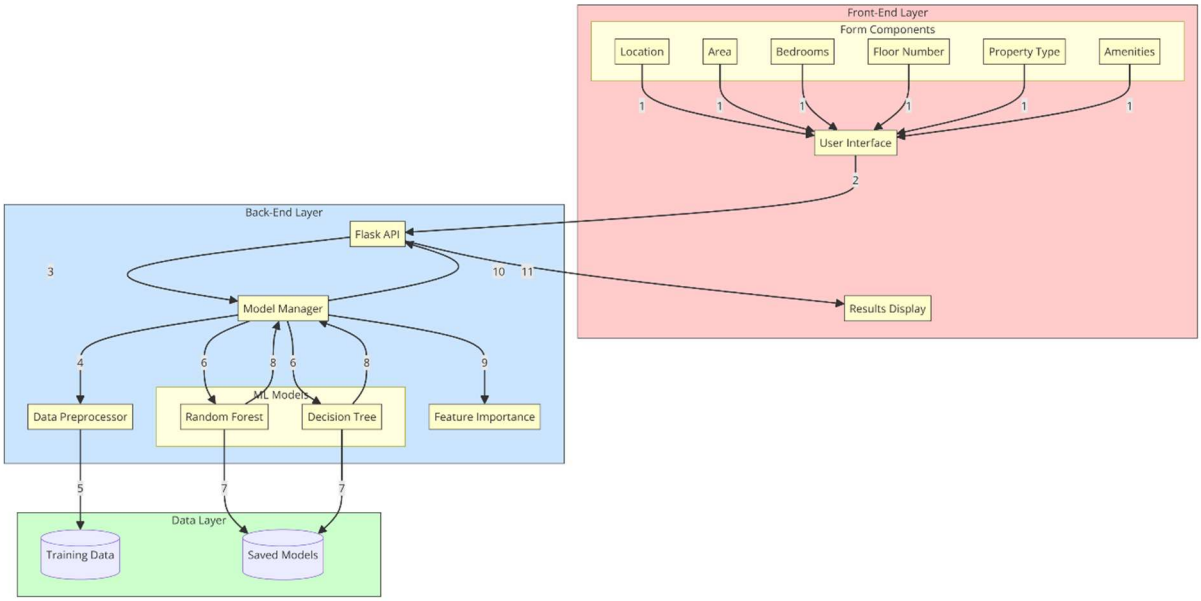
The system implements robust error handling at each stage. Input validation ensures data quality, while the preprocessing stage handles missing or invalid values gracefully. The Model Manager implements fallback mechanisms if one model fails, ensuring system reliability. The Data Layer maintains separation between training data and model storage, allowing for independent updates and maintenance. The color-coded communication diagram illustrates these relationships: purple for front-end components, blue for backend processing, and green for data storage, making the system architecture easily understandable.



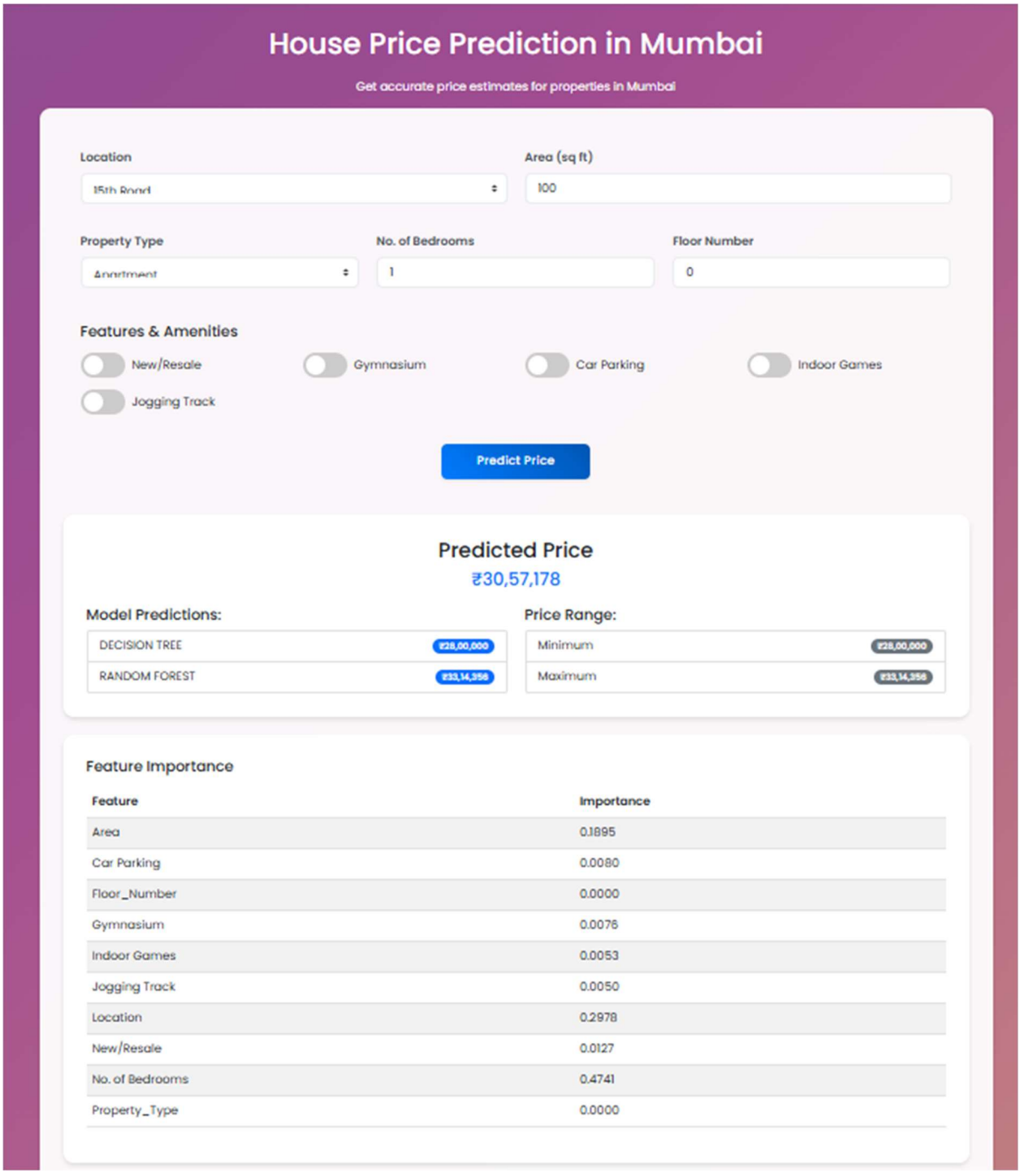
Communication Diagram:

It illustrates the interactions between the system's many parts, particularly the data and message flow between the Flask API, the user interface, and the backend model management and

processing components. The message flow and sequence that enable various system components to interact and communicate are the main emphasis of this kind of diagram.



UI for the project



Github:

https://github.com/ks-tharun-14/CSCE5214_SDAI_PROJECT_GROUP21