# Chapter6 Module Designs

# 1 User Authentication Module

**6.1.1 Purpose and Functionality**

The User Authentication Module handles user registration, login, and identity verification. It will:  
• Enable new users to register securely  
• Authenticate existing users via username and password  
• Maintain session control and logout functionality

**6.1.2 Module Components**

1. **Registration Component**• Functionality: Registers new users into the system  
   • Interfaces:

– create\_user\_with\_verification(db , user\_in )

– create\_user(db , user\_in )

1. **Login Component**• Functionality: Authenticates user credentials and initiates sessions  
   • Interfaces:

– get\_user(db , id)

– get\_user\_by\_email(db , email)

– get\_user\_by\_username(db , username)

– authenticate\_user(db , username\_or\_email , password )

**6.1.3 Data Flow**

1. User submits registration/login info via frontend
2. Backend verifies and interacts with user database
3. Authentication token is issued and stored in session
4. Token is used for secure route access and operations

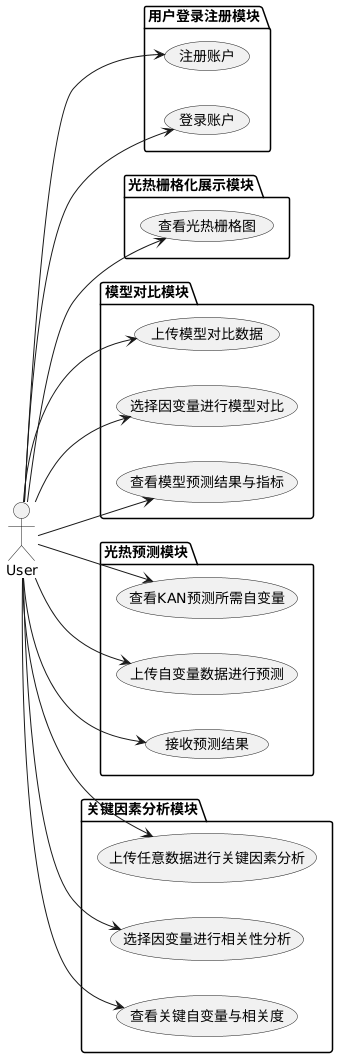


Figure 6.1: Use Case Diagram 1

**2 Raster Visualization Module**

**6.2.1 Purpose and Functionality**

The Raster Visualization Module displays Shanghai's gridded thermal or light data on an interactive map. It allows selection from:  
• Daytime Heat  
• Nighttime Heat  
• Nighttime Light Intensity

**6.2.2 Module Components**

1. **Data Selector**  
   • Functionality: Allows users to choose the raster data category  
   • Interfaces:  
    – selection()
2. **Map Renderer**  
   • Functionality: Renders selected raster data on the Shanghai map grid  
   • Interfaces:  
    –updateCharts(variable)
3. **Data Fetcher**  
   • Functionality: Communicates with backend to fetch raster data  
   • Interfaces:  
    – load\_data\_by\_target(target , file\_path)

**6.2.3 Data Flow**

1. User selects a data category
2. Frontend requests corresponding raster data from backend
3. Raster data is rendered as a heatmap on the grid-based map

**3 Model Comparison Module**

**6.3.1 Purpose and Functionality**

The Model Comparison Module allows users to upload data and compare predictive performance of MLP, RF, and KAN models on selected dependent variables. It will:  
• Accept Excel files in a defined format  
• Let users select one dependent variable (from Daytime Heat, Nighttime Heat, Nighttime Light)  
• Generate predictions using MLP, RF, and KAN  
• Display MSE and R² comparisons

**6.3.2 Module Components**

1. **File Upload Interface**  
   • Functionality: Uploads dataset to server  
   • Interfaces:  
    – handleFile(file)
2. **Model Runner**  
   • Functionality: Executes three models on selected target  
   • Interfaces:  
    – analyze\_and\_predict(df , target\_dependent\_var)
3. **Metrics Comparator**  
   • Functionality: Computes and displays evaluation metrics  
   • Interfaces:  
    –updateCharts(variable)

**6.3.3 Data Flow**

1. User uploads Excel file and selects a dependent variable
2. Backend preprocesses data and runs MLP, RF, KAN
3. Model outputs and metrics are returned and visualized

**4 Light-Heat Prediction Module**

**6.4.1 Purpose and Functionality**

This module uses KAN to predict any of the three target variables based on important features. It will:  
• Show static lists of key input features and their importance per target  
• Allow uploading of new data for prediction  
• Return predicted values using KAN

**6.4.2 Module Components**

1. **Feature Importance Viewer**  
   • Functionality: Displays top features with correlation for each target  
   • Interfaces:  
    –Display in static form
2. **Prediction Interface**  
   • Functionality: Uploads Excel and receives KAN prediction  
   • Interfaces:  
    –predict\_from\_excel(df , target\_dependent\_var)

**6.4.3 Data Flow**

1. User selects target and views required features
2. User uploads new data containing those features
3. Backend performs prediction and sends back results

**5 Key Factor Analysis Module**

**6.5.1 Purpose and Functionality**

The Key Factor Analysis Module identifies and displays the most relevant independent variables related to a user-selected dependent variable. It will:  
• Accept any Excel file with multiple columns  
• Allow user to select one dependent variable  
• Analyze variable correlations  
• Display top influencing factors and correlation scores

**6.5.2 Module Components**

1. **Data Input and Target Selection**  
   • Functionality: Accepts Excel and selects dependent variable  
   • Interfaces:  
    – handleFile(file)  
    – selection()
2. **Correlation Analyzer**  
   • Functionality: Identifies significant predictors for the target  
   • Interfaces:  
    – analyze\_key\_features\_from\_df(Df , Target , n\_epochs , hidden\_dim , Grid , K , lr)
3. **Result Display**  
   • Functionality: Displays key variables and their strengths  
   • Interfaces:  
    – updateCharts(data)

**6.5.3 Data Flow**

1. User uploads data and chooses a dependent variable
2. Backend computes feature correlations
3. Results are ranked and shown with visuals