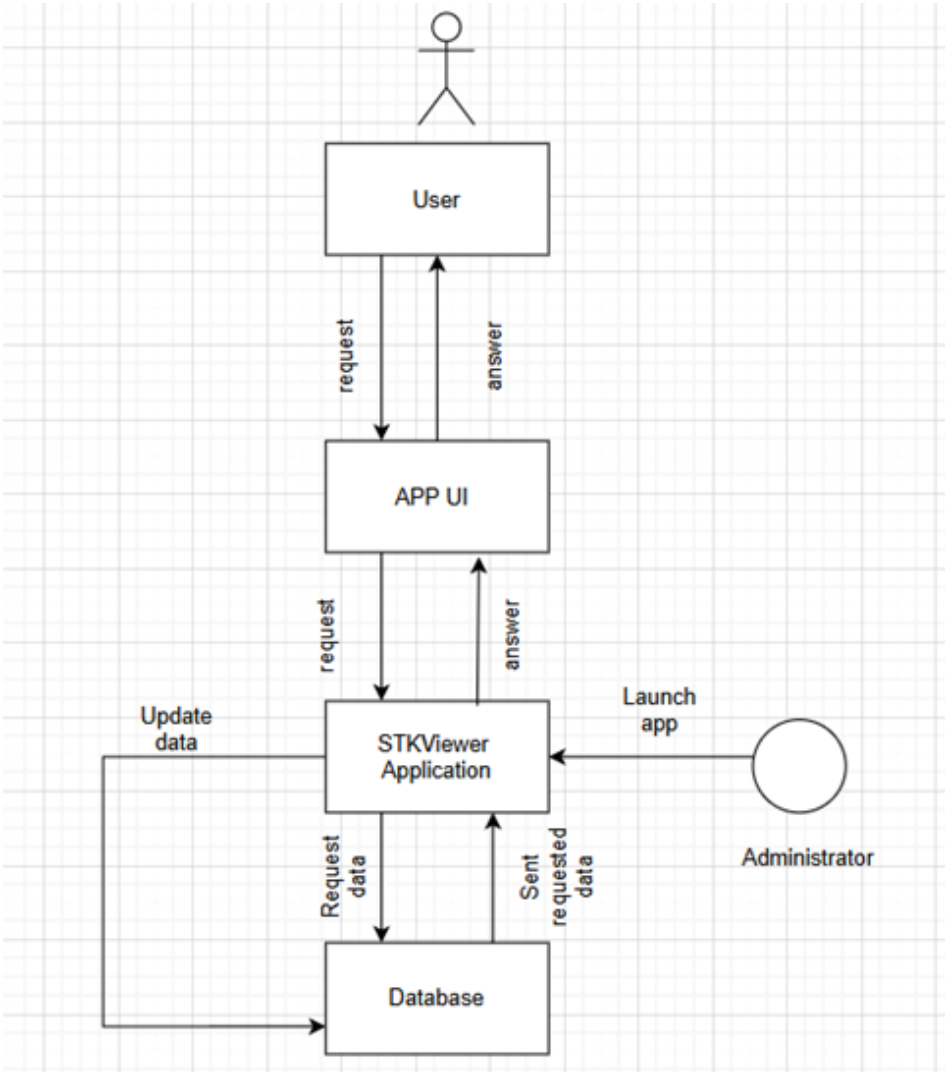
# Conceptual architecture

Through the conceptual architecture, we demonstrate the connection between the components of the application and the flow of data. The conceptual architecture is created based on the analysis of user requirements. Below is the diagram developed according to the conceptual model.




## Core Components:

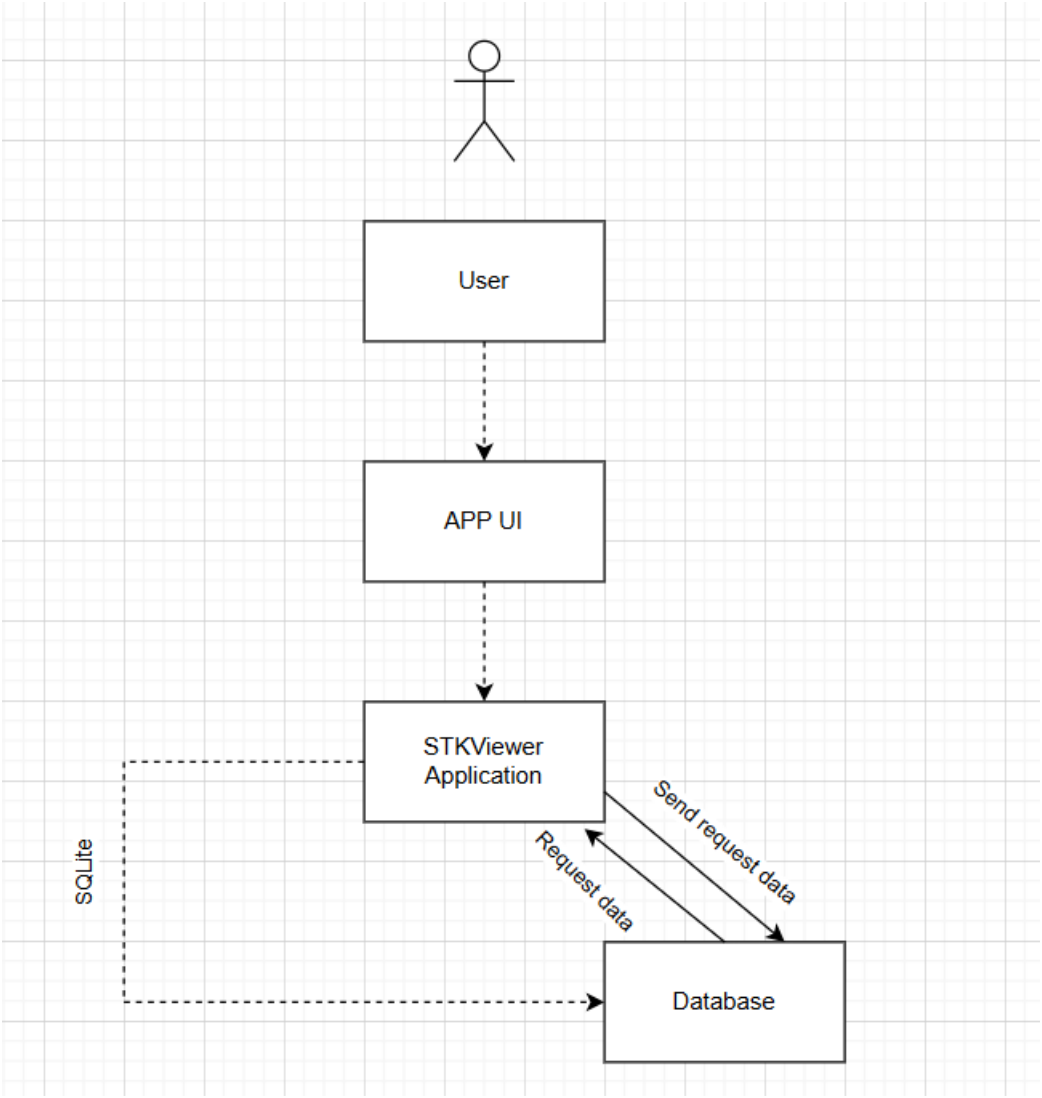
- **User:** Initiates requests through the APP UI and receives responses.
- **APP UI:** Acts as an interface between the user and the system, forwarding requests to the core application and displaying returned data.
- **STKViewer Application:** The central processing unit that handles user requests, communicates with the database, and manages data updates. It ensures accurate responses and maintains system logic.
- **Database:** Stores and retrieves critical information. It processes data requests from the STKViewer Application, ensuring consistency and integrity.
- **Administrator:** The user which enables the program.

# Execution architecture

This diagram illustrates the interaction between key components in a web application. The **User** sends HTTP requests through the **APP UI**, which acts as the intermediary between the user and the backend. The **APP UI** communicates with the **STKViewer Application** using JavaScript, forwarding requests for processing.

The **STKViewer Application** handles the core logic and interacts with the **Database** through SQL commands to fetch or update data. The data is then returned to the **APP UI**, which displays it to the user.

This architecture ensures a clear separation of concerns—user interface, application logic, and data storage—resulting in an efficient, scalable, and maintainable system.



# Implementation architecture

This implementation architecture diagram illustrates the interaction between core components of the system:
User: Interacts with the APP UI through HTTP requests.
APP UI: The front-end interface, built with JavaScript, communicates with the STKViewer Application to send requests and display responses.
STKViewer Application: Handles logic and processes data queries using SQLite.
Database: Stores and retrieves data upon request from the STKViewer Application via SQLite operations.
The architecture ensures modularity, with clear roles for each component, supporting scalability and efficient data handling.