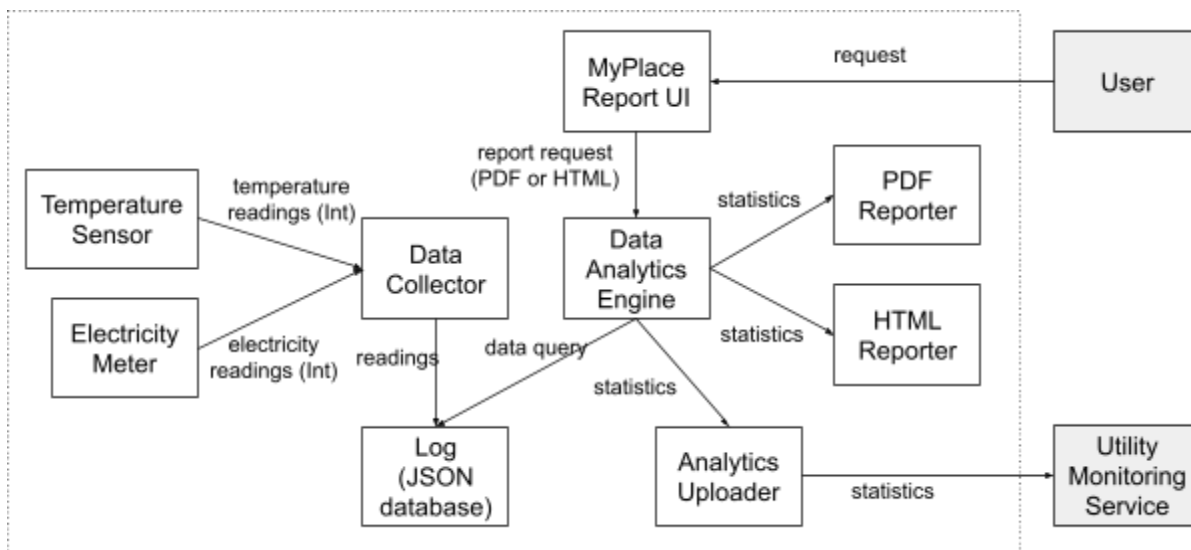


Andrew IDs:

MyPlace is an Internet-of-the-Things (IoT) system that periodically collects data from different types of sensors throughout a smart building (temperature sensors, electricity meters, etc.) and provides the building owner with various statistics about the collected information (e.g., temperature trends over the week, average and maximum electricity usage, etc.). Below is a component diagram that shows the current design of the system; User and Utility Monitoring Service are domain entities that interact with the system.



The responsibilities of the components are:

- **Temperature sensor, electricity meter:** Generate temperature/electricity readings at certain periods and send them to the data collector.
- **Data collector:** Stores the readings received from the sensors/meters into a log database.
- **Log:** Stores all of the past readings as a database.
- **MyPlace report UI:** The frontend for the system. Receives requests from the user to generate statistics reports either in form of (1) HTML, to be immediately displayed on a browser or (2) PDF, to be downloaded and stored on the user's device for archival purposes.
- **Data analytics engine:** Based on the user request from the UI, queries the log database for necessary data and computes the requested statistics (e.g., "average electricity usage for the last 6 months"). Then, the engine passes the statistics to either the PDF or HTML reporter (depending on the type of request) and returns the generated report back to the UI. In addition, the engine periodically computes a predefined set of statistics (e.g., "weekly electricity consumption") and sends it to the analytics uploader.
- **Analytics uploader:** Uploads the statistics computed by the analytics engine to the external utility monitoring service.
- **PDF/HTML reporter:** Given the statistics from the analytics engine, produce a report either in PDF or HTML format.

Q1. What are possible changes that may occur in the **MyPlace** system? Describe the impact of each change on the rest of the system.

Q2. Does the system violate any of the following principles? If so, how would you redesign the system to fix those violations and improve the changeability of the system?

- Information hiding
- Single-responsibility
- Interface segregation
- Dependency inversion