Online monitoring system of city and intercity buses

In the era of information technology, the use of cloud services has emerged as a pioneering and innovative solution across various industries. One notable application of cloud technology is in the field of public transportation, where the implementation of an online monitoring system for urban and intercity buses has brought about a significant transformation in the operation and management of this system. This monitoring system is designed and implemented based on cloud services. It uses the internet and a connection to a control center to enable real-time monitoring and supervision of all buses in the public transportation network. Comprehensive information, including the exact location of buses, passenger numbers, fuel consumption, driver ratings, and vehicle technical status, is provided to the control center.

The use of cloud services in this system offers substantial benefits to the public transportation sector. These services provide scalability, easy access, and improved information security to managers, drivers, and passengers. Additionally, cloud services contribute to reducing hardware and software costs, enhancing communication and collaboration among system members, and facilitating quick and accurate decision-making. Given the importance and impact of the online bus monitoring system on improving the quality and efficiency of public transportation, this study will examine the advantages and features that this cloud-based system offers. It will also explore the concept and characteristics of the cloud domain.

This research aims to provide a more precise introduction and explanation of the achievements and benefits of using cloud services in the public transportation sector, thereby increasing general understanding of this innovative technology. The system is implemented using Python programming language due to its simplicity and powerful programming capabilities, making it suitable for developing and implementing complex systems. The deployment of this system leverages Infrastructure as a Service (IaaS) from cloud service providers. These services include server infrastructure, networking, and storage in the form of virtual machines, offering scalability and flexibility. For those interested in continuing this project and further developing it, a potential next step could be the addition of a dedicated mobile application. This mobile application would enable users to easily access the online monitoring system's features through their mobile devices, allowing them to stay connected with public transportation information at all times and locations.

Architecture and Implementation

The architecture of this project consists of three main components:

1. Server
2. Buses
3. Users

1. Buses

The bus acts as a Publisher client that continuously sends its data to the server. This data includes precise geographical location, the number of passengers on board, the driver's name, the driver's image, and the timestamp of the last information sent. Given that in a real-world scenario, this client would run on an embedded system and the absence of a need to implement an embedded system in the cloud computing course, a piece of code named busClient.py has been implemented in Python to simulate the operation of this part of the system.

2. Users

The user acts as a Subscriber client that receives the necessary services from the server. These services are typically accessed via an application on Android and iOS smartphones. However, due to time constraints for project delivery, a piece of code named appClient.py has been implemented in Python to simulate the operation of this part of the system.

3. Server

The server communicates with the buses and users using sockets. The server code, named server.py, has been implemented in Python. The server in this project includes several components:

- **Executable Program Code**: Runs on the Ferdowsi Cloud Virtual Machine.
- **Databases**: Includes a local relational database (SQLite on Ferdowsi Cloud) and a non relational cloud database (MongoDB).
- **File Storage**: Uses Arvan Cloud Object Storage.
- **APIs**: Provides necessary interfaces for communication and data exchange.