Data Architecture Challenge

We are managing parking lots that a client can check with a mobile app. An app can tell the drive if a parking is full or not. On entering/leaving the parking a client can scan QR/NFC code on entrance machines and the cost must be automatically charged when leaving parking. We are interested in monitoring when the parking is full or empty to modify prices accordingly. We also would like to create a predictive model that learns when a client is going to the parking to send him a push message informing how many places are left or if the parking is full.

1. What tracking events would you propose? What data model for event analysis? What technologies?

2. How would you design the Backend system? What data model for the Operational system? What technologies?

3. Explain how to combine the operational architecture with the analytical one?

4. Could you propose a process to manage the development lifecycle? And the test and deployment automation?

Solution:

1. For tracking events we could spark-streaming with kafka to know the current status of the parking lot. Time Series would be much more apt considering this scenario.

In terms of tech-stack: Spark-Streaming with or without Kafka would be nice.

For storing this data, we could use the Postgres database and Redshift as well.

1. The backend system would be a Postgres database for storing the entries based on the events. For streaming we could leverage spark-streaming with a suitable window.
2. We could use Redshift for analytical usage while postgres for operational usage.
3. Jira process with a 2 weeks sprint would be feasible for the development lifecycle.

For testing :- preferably unit-test cases and integration test cases would be much suitable.

For deployment :- build the entire pipeline with all the relevant dependencies using docker and kubernetes.

