Designing an AWS Cloud-Based Solution for a Parking Management Application

FIT5225 2024 SM1

1 Synopsis and Background

PARKED, a parking management operator company, has decided to migrate its on-premise parking management application to Amazon Web Services (AWS) and has hired you as a Cloud Solution Architect/Consultant to design their cloud solution. As an AWS solution architect, you will be responsible for designing a cloud-based solution to efficiently monitor and regulate parking activities.

A Parking Management System equipped with license plate recognition technology. At the entrance, it monitors vehicles entering the parking area and issues tickets based on the duration of parking at the exit. Here's how it operates:

- **Entry Registration**: When a vehicle enters the parking area, the system's cameras capture images of its license plate. The system records the entry time and associates it with the vehicle's license plate number.
- **Parking Duration Monitoring**: As the vehicle remains parked, the system continuously tracks the duration of its stay.
- **Overstay Detection**: If a vehicle exceeds the allowed free parking duration, the system identifies it as a violator, alerts the mobile apps of the vehicle owners, and starts charging.
- **Plate Verification**: The system compares the license plate captured at the exit with the entry records in its database to verify the vehicle's duration of stay.
- **Exit Process**: When the vehicle attempts to leave the parking area, the system's cameras once again capture images of its license plate, issue a ticket and opens the gate after the payment is made.
- **Ticket Issuance**: If the vehicle has exceeded the permitted free parking duration, the system generates an electronic ticket at the exit point. The ticket includes details such as the license plate number, entry and exit times, and charging fee sent to the mobile app.
- **Alerts and Notifications**: Real-time alerts and notifications are sent to parking attendants. This enables prompt action to be taken to address parking overtime stay.
- **Reporting and Analytics**: The system generates reports and analytics based on parking data collected over time. These insights can be used to optimize parking space utilization, identify patterns in parking utilization, and improve overall parking management strategies.

Your solution will leverage AWS services and your knowledge gained in the FIT5225 unit to enable PARKED to collect fees from drivers using their mobile application system. You may spend some time researching about how automated parking systems work.

2 Technical aspects

Please prepare your report based on the following guidelines:

- Include an architectural diagram of your solution using AWS Architecture Icons. You can find more information and download the icons from the official AWS website: `https://aws.amazon.com/architecture/icons/`. Your architecture diagram should illustrate the various components of your solution, including the AWS services used, and how they interact with each other. You can use Microsoft PowerPoint or any other tools like this: `https://www.draw.io/index.html`.
- In your design explanation, ensure to consider and address key aspects such as **Scalability**, **Security**, **Failure Handling** (including availability and reliability) and **cost-effectiveness**. To do so, you may need to make reasonable assumptions about the system's requirements, constraints, and limitations, but ensure that each assumption is justifiable based on the information provided. Explain how each assumption influenced your design decisions, with a particular focus on these key aspects.
- Provide clear and concise explanations that demonstrate your understanding of AWS services, including why you selected specific AWS services, what specific type of that service was used, and which settings were considered. Your explanation should emphasize on how your application works and how various components of the architecture interact with each other.
- The quality of your design will be evaluated based on its **completeness**, **correctness**, and **clarity**. Make sure to include all the relevant components and AWS services required to implement your solution. Use clear and consistent labelling and notation in your architecture diagram. Finally, make sure your design is easy to read and understand.
- For this assessment, you are permitted to utilize **ChatGPT** and Generative AI large language models, if you wish. However, please be sure to provide a clear explanation of how and where you have incorporated ChatGPT into your work; also, include a list of main prompts you have used (You can attach this as an appendix to your report).
- You are allowed to make your own assumptions and imagine how the parking management system should work based on the explanation provided in this assignment description. Your solution should focus on the tasks of reading and storing vehicles' plates and images, and issuing charging bills and tickets for drivers. You may assume that your system has access to a relational database (such as Aurora RDS) containing all necessary information regarding drivers and their vehicles.

3 Submission

You need to submit **a report** via Moodle:

- The report must be in PDF format.
- The report should not exceed 1500 words.
- Only **ONE** diagram should be included in the report apart from the text.
- A cover sheet must be included in the report, containing all necessary information such as student name, ID, unit, tutors, etc.