SCHOOL OF COMPUTER SCIENCES

ACADEMIC SESSION: 2025/2026

CSE443 REAL-TIME SOFTWARE ENGINEERING

INDIVIDUAL ASSIGNMENT 1

LECTURER'S NAME : DR. FAZILAH OTHMAN

Student Name	Matric Number
Nor Shafiqa Atirah binti Mohd Zaidi	164211

Submission date:

30th April 2026

Assignment 1 – Marking Rubric

	Poor (0-30%) D+, D, D-, F	Average (31-60%) C+, C, C-,	Good (61-75%) B+, B, B-	Excellent (76-100%) A, A –	Marks
Section 1: Introduction to the Real-time System	Wrong choice of real-time system. The introduction is out of topic and lacks explanation of the industry domain or importance of real-time behavior.	The choice of real-time system is partially relevant. A fair description of the industry domain and importance of the system is provided.	The choice of real-time system is appropriate. A moderately good description of the industry domain and importance of the system is provided.	The choice of real-time system is relevant. An excellent description of the industry domain, importance of real-time behavior, and system classification is provided.	/100 /10
Section 2: System Requirements Analysis and Specification	No use case diagram or wrong diagram illustration. Functional and non-functional requirements are missing or incorrectly described.	A use case diagram is provided but contains significant errors or incomplete explanation. Functional and non-functional requirements are partially identified with weak descriptions.	A use case diagram with explanation is provided with minor errors. Functional and non-functional requirements are reasonably described with minor corrections.	A clear and well-structured use case diagram with explanation is provided. Functional and non-functional requirements are accurately identified and clearly described.	/100 /40
Section 3: System Design Review	No system architecture description is provided or the description is incorrect. System components are not identified and Functional Flow Diagram is missing or incorrect.	A basic description of system architecture is provided but with limited clarity. System components are partially identified and the Functional Flow Diagram contains major errors.	A reasonably clear description of system architecture is provided. Main system components and interactions are identified with minor corrections. Functional Flow Diagram illustrates the system process with minor errors.	A clear and well-structured system architecture description is provided. Main system components and interactions are clearly described. Functional Flow Diagram accurately illustrates system inputs, processes, decisions and outputs.	/100 /20
Section 4: Suggestion & Improvement	No suggestion provided or suggestions are irrelevant to the system.	Only one improvement suggestion is provided or the explanation is weak.	Two improvement suggestions are provided with reasonable justification.	Two well-justified improvement suggestions are provided with clear justifications and relevance to system performance or reliability.	/100 /20
Reference & Writing Style	No citations or references are provided. Poor writing structure and unclear presentation.	Less than five references are provided or citation style is inconsistent. Writing style is moderately unclear.	At least five references are provided with acceptable citation style. Writing is generally clear and structured.	Five or more references are provided with correct citation style. Writing is clear, professional, and well structured.	/100 /10
					/100

<u>Table of Contents</u>

1.0. Introduction to the Real-Time System

The Smart Parking Monitoring System is a real-time solution designed to enhance the efficiency of parking space management through modern technology. It integrates IoT sensors, cameras, and software applications to continuously monitor parking availability and provide instant updates to driver. Through mobile and web – based platform, drivers can easily view availability parking slots and reduce search time. By combining sensor data with camera detection, the system improves accuracy and makes it more convenient for drivers [1].

This system is widely applied in the transportation and smart city domain, particularly in urban centres, shopping malls, airports, and commercial complexes where parking demand is consistently high. Real-time operation is important because parking information needs to be updated immediately whenever vehicles enter or leave. With fast communication between devices, the system provides quick and accurate updates. This helps drivers make faster decisions, reduces traffic caused by searching for parking, and improves congestion in urban areas [3].

From a systems perspective, such applications are classified based on timing constraints and criticality levels, where systems may range from hard real-time to soft real-time depending on the impact of missed deadlines [6]. In this case, the system is classified as a soft real-time system because minor delays are acceptable without causing serious failure. Additionally, it is an embedded system as it relies on IoT-based sensors and cameras integrated into the parking infrastructure to support continuous monitoring and control.

2.0. System Requirements Analysis and Specification

2.1. Use Case Diagram

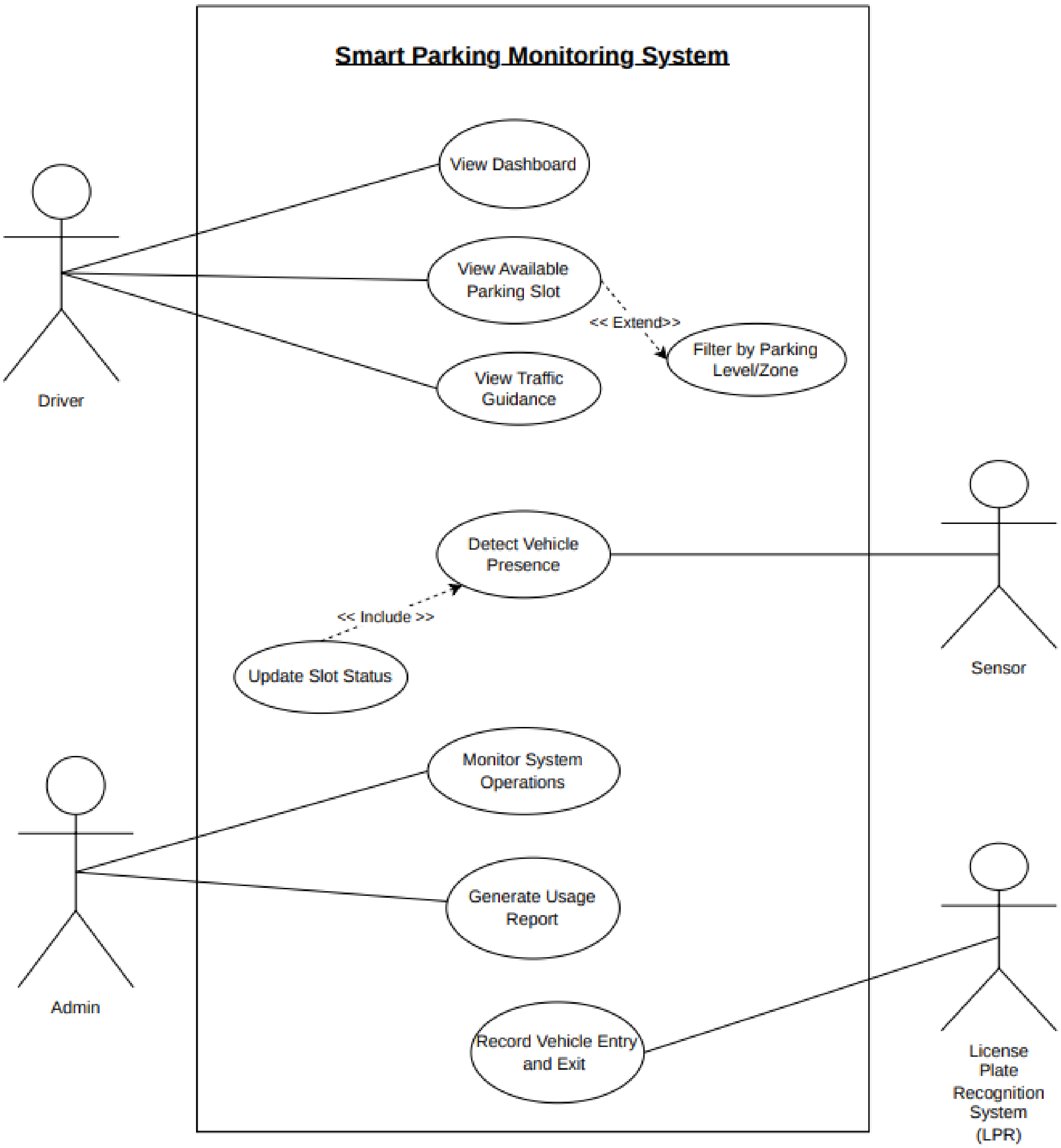

Figure 2.1.1: Use Case Diagram of "Smart Parking Monitoring System"

The Smart Parking Monitoring System use case diagram (Figure 2.1.1) illustrates the interaction between its main actors within a defined system boundary. The driver is the primary actor can access the system through a mobile or web interface. The driver can view a dashboard that presents key information such as available and occupied parking slot. In addition, the driver is able to view available parking slots and apply filters by parking level or zone, allowing for quicker and more efficient slot selection. The system also provides traffic guidance to direct drivers toward less congested areas or the nearest available parking slot, helping to minimise search time and reduce internal congestion.

The Sensor is responsible for detecting vehicle presence in each parking slot. This function is directly associated with updating the slot status, ensuring that the information displayed to drivers remains accurate and up to date. To enhance system capability, cameras equipped with License Plate Recognition (LPR) are used to capture vehicle entry and exit details along with corresponding timestamps. This supports better monitoring of vehicle flow, improves traceability, and adds an additional layer of security within the parking facility.

The admin actor can manage and oversees system operations. This includes monitoring system performance, checking sensor functionality, and ensuring overall reliability. The admin is also able to generate usage reports, which provide insights into occupancy patterns, peak usage periods, and parking turnover. As a result, these interactions demonstrate a well-integrated system designed to support efficient parking management and enhance driver experience.

2.2. Functional Requirements

Item	**Description**
Function ID	FR1
Function Name	View Available Parking Slot
Function Description	The system shall allow drivers to view real-time parking slot availability through a mobile or web interface.
Triggering Event	Driver opens the application or refreshes the parking map.
System Inputs	Data from parking slot sensors indicating occupancy status (available or occupied).
System Outputs	Drivers can view updated visual display of parking slots showing availability status with clear indicators.
Timing Constraint	The system shall update and display changes in parking slot availability within 3 seconds after any change in slot status is detected.

Table 2.2.1: Functional Requirements of "View Available Parking Slot"

Item	**Description**
Function ID	FR2
Function Name	Update Parking Slot Status
Function Description	The system shall automatically update the status of each parking slot (available or occupied) whenever a vehicle enters or leaves a parking slot.
Triggering Event	A vehicle enters or exits a parking slot, detected by IoT-based sensors.

System Inputs	Real-time sensor data indicating the presence or absence of a vehicle in each parking slot.
System Outputs	Updated slot status stored in the system database and displayed on the driver's dashboard and parking map.
Timing Constraint	The system shall transmit sensor data to the database within 3 seconds of detecting any change in parking slot occupancy caused by a vehicle entering or exiting.

Table 2.2.2: Functional Requirements of "Update Parking Slot Status"

2.3. Non - Functional Requirements

Requirement Type	**Description**
Performance	The system must update parking availability data within 3 -5 seconds and support multiple concurrent users (drivers) without performance degradation, especially in high-traffic areas.
Reliability	The system must maintain at least 99% uptime and ensure accurate synchronization between sensors, LPR, and database to avoid incorrect parking records or slot allocation errors.
Environmental / Operational Constraint	The system must operate effectively in indoor and outdoor environments, including low light, rain, and high traffic conditions, and maintain stable network connectivity across distributed components.

Table 2.3.1: "System Non – Functional Requirements"

3.0. System Design Review

This system uses a layered IoT-based distributed architecture consisting of sensing, processing, and application layers to support real-time smart parking management. The sensing layer consists of IoT sensors and cameras installed at each parking slot to detect vehicle presence and capture license plate information in real time. It also includes License Plate Recognition (LPR) using cameras at entry and exit points to identify vehicles and support tracking. These devices continuously collect occupancy data and send it for processing. This approach is similar to modern smart parking systems that use wireless sensor networks (WSN) and RFID technology for accurate vehicle detection and identification [2].

The processing layer uses edge computing devices together with a central server. It receives data from sensors, processes slot availability, and updates the central database to ensure real-time parking information remains accurate. By processing data closer to the source, edge computing reduces communication delay and improves response time while reducing the load on the central server. This distributed approach is widely used in intelligent transportation systems, where computation is shifted to local gateways to improve scalability and avoid system congestion [4].

The application layer allows drivers to interact with the system through mobile and web interfaces. Drivers can view available parking spaces, locate empty slots, and receive navigation guidance. Administrators can monitor system performance, check sensor status, and generate reports for analysis. The system is deployed in multi-storey parking facilities such as shopping malls, airports, hospitals, and commercial complexes where parking demand is high. Sensors and cameras are installed at each parking bay as well as at entry and exit points to ensure complete coverage of the parking area. As a result, the system relies on continuous data exchange between its layers, enabling real-time updates, high scalability, and efficient parking management.

3.1. Functional Flow Diagram

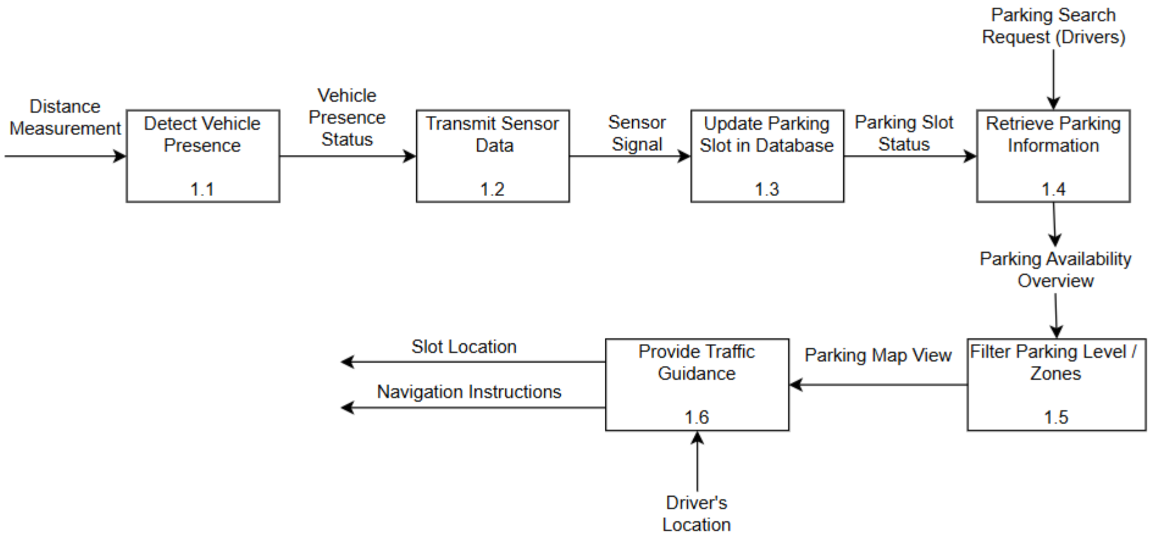

Figure 3.1: "Functional Flow Diagram"

The above diagram (Figure 3.1) illustrates how sensor data is processed and transformed into guidance for drivers. The process begins with distance measurement, where sensors detect the presence of a vehicle in a parking slot. This data is analysed to determine the vehicle presence status (occupied or available). The status is then transmitted as a sensor signal to the central system, where the parking database is updated to ensure accurate and real-time slot information.

When a driver submits a parking search request, the system retrieves parking information and generates a parking availability overview. The data is then filtered based on the selected parking level or zone, producing a parking map view. Finally, the system uses the driver's location together with the filtered data to generate navigation instructions, guiding the driver to an available parking slot efficiently.

4.0. Suggestions and Improvements

- **Suggestion 1: Improve System Reliability through Fault Tolerance**

  One improvement is to increase the reliability of the Smart Parking Monitoring System by adding fault tolerance mechanisms. At present, the system relies heavily on sensor nodes and gateways to provide real-time parking information. If a sensor fails or network communication is interrupted, the system may display incorrect or outdated data. To reduce this risk, backup sensors can be installed in key parking areas, and alternative communication routes such as mesh networking or secondary gateways can be used. This ensures that data can still be transmitted even when one component fails. In addition, a simple self-monitoring feature can be added so that faulty sensors are detected early and maintenance alerts are triggered. This improves system stability and reduces downtime, ensuring more reliable real-time parking information.

- **Suggestion 2: Enhance Performance using Edge Processing**

  Another improvement focuses on system performance. Currently, all sensor data is sent directly to the central server, which can create delays during peak traffic conditions. This can be improved by using edge computing at gateway level, where data is processed and filtered before being sent to the main server. This reduces network traffic and allows faster updates for users. In addition, switching from continuous data transmission to an event-based approach will improve efficiency, as only changes in parking status are sent. This reduces unnecessary communication and improves response time. Overall, this makes the system faster and more efficient, especially in busy urban environments.

References:

[1] Abdelrahman Osman Elfaki, Wassim Messoudi, Anas Bushnag, Shakour Abuzneid, and Tareq Alhmiedat, “A Smart Real-Time Parking Control and Monitoring System,” *Sensors*, vol. 23, no. 24, pp. 9741–9741, Dec. 2023, doi: https://doi.org/10.3390/s23249741.

[2] A. Hilmani, A. Maizate, and L. Hassouni, “Designing and Managing a Smart Parking System Using Wireless Sensor Networks,” *Journal of Sensor and Actuator Networks*, vol. 7, no. 2, p. 24, Jun. 2018, doi: https://doi.org/10.3390/jsan7020024.

[3] M. A. Ala’anzy, A. Abilakim, R. Zhanuzak, and L. Li, “Real time smart parking system based on IoT and fog computing evaluated through a practical case study,” *Scientific Reports*, vol. 15, no. 1, Sep. 2025, doi: https://doi.org/10.1038/s41598-025-15507-6.

[4] M. Alam *et al.*, “Real-Time Smart Parking Systems Integration in Distributed ITS for Smart Cities,” *Journal of Advanced Transportation*, vol. 2018, pp. 1–13, Oct. 2018, doi: https://doi.org/10.1155/2018/1485652.

[5] What is Functional Flow Block Diagram (FFBD)?,” *online.visual-paradigm.com*. https://online.visual-paradigm.com/resources/functional-flow-block-diagram/guides/what-is-functional-flow-block-diagram/

[6] J. Cooling, *The complete edition – Software Engineering for Real-Time Systems*. Packt Publishing Ltd., 2019.