

## Customer Problem Statement

**Problem Statement:** Describe the problem your customers encounter and their expectations of how a software system would help. Describe the problem like a narrative about your project in the role of a customer instead of a developer. The problem description should be **based on the previously submitted project proposal**. You can revise and improve it based on the feedback of the proposal if necessary. (2 points)

example:

"A **parking lot** is a designated area for parking vehicles and is a feature found in almost all popular venues such as shopping malls, sports stadiums, offices, etc. In a parking lot, there are a fixed number of parking spots available for different types of vehicles. Each of these spots is charged according to the time the vehicle has been parked in the parking lot. The parking time is tracked with a parking ticket issued to the vehicle at the entrance of the parking lot. Once the vehicle is ready to exit, it can either pay at the automated exit panel or to the parking agent at the exit using a card or cash payment method."

**Glossary of Terms:** Define important terms to make them consistent in the system specification. Add pictures or graphics to explain your idea. Avoid defining computer science terms such as network protocol, thread, client/server, and user interface. Only define new terms that you introduce into your system. (2 points)

**List System Requirements which would include Functional Requirements, Non-Functional requirements and User Interface Requirements.**

**System Requirements**

- Functional Requirements (6 points)

Extract and list the requirements from the problem statement in the table shown below.

| No.  | Priority Weight                        | Description                      |
|--|--|----------------------------------|
| REQ-1 (show the unique label of the requirement) | The priority weight of the requirement | Describe the requirement briefly |

Example:

**Functional Requirements**

| No.   | Priority Weight | Description  |
|-------|-----------------|--|
| REQ-1 | High            | The parking lot should have the capacity to park 30,000 vehicles   |
| REQ-2 | High            | The four different types of parking spots are handicapped, compact, large, and motorcycle.                                       |
| REQ-3 | Low             | The parking lot should have multiple entrances and exit points.  |
| REQ-4 | High            | Four types of vehicles should be allowed to park in the parking lot, which are as follows: car, truck, van, motorcycle           |
| REQ-5 | High            | The parking lot should have a display board that shows free parking spots for each parking spot type                             |
| REQ-6 | High            | The system should not allow more vehicles in the parking lot if the maximum capacity (30,000) is reached.                        |
| REQ-7 | Medium          | If the parking lot is completely occupied, the system should show a message on the entrance and on the parking lot display board |
| REQ-8 | Medium          | Customers should be able to collect a parking ticket from the entrance and pay at the exit.                                      |

|        |        |   |
|--------|--------|---|
| REQ-9  | High   | The customer can pay for the ticket either with an automated exit panel or pay the parking agent at the exit. |
| REQ-10 | Medium | The payment should be calculated at an hourly rate  |
| REQ-11 | Medium | Payment can be made using either a credit/debit card or cash  |

- **Nonfunctional Requirements (2 points)**

Example : Nonfunctional Requirements

| No.   | Priority Weight | Description  |
|-------|-----------------|--|
| REQ-1 | High            | The system shall handle peak loads during events or rush hours without significant performance degradation. It should process payment transactions swiftly and accurately. |
| REQ-2 | High            | The system shall ensure the security of payment transactions and vehicle data. It should comply with relevant data protection regulations (e.g., GDPR, PCI-DSS).           |
| REQ-3 | High            | The system shall integrate with existing security systems (CCTV cameras), financial systems for payment processing, and management tools for reporting.                    |
| REQ-4 | Low             | The system architecture shall support future expansion of parking spots and integration with advanced technologies (e.g., smart parking sensors).                          |

**User Interface Requirements:** List, prioritize, and describe the user interface requirements and add a graphic for each requirement. Simply use sketches for this step. (6 points)

**Plan of Work (2 points)**

Please use this section to report the progress of your project. We can put the plan of the work of the proposal here and put a mark on the item which you have finished. For example, we focus on setting up the development environment this week, so you could describe how's this process going.

**When to start coding**

**It would be helpful to start implementing some features of your project immediately.**